(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2007-250180
(P2007-250180A)
(43)公開日　平成19年9月27日(2007.9.27)

(51)Int.Cl.		F I			テーマコード（参考）
G11B 20/10	(2006.01)	G11B	20/10	A	5D044
		G11B	20/10	301Z	

審査請求 有　請求項の数 4　ＯＬ　（全 34 頁）

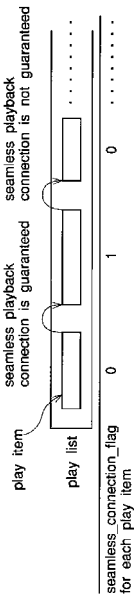
(21)出願番号　特願2007-138560 (P2007-138560)
(22)出願日　平成19年5月25日 (2007.5.25)
(62)分割の表示　特願平10-120393の分割
原出願日　平成10年4月30日 (1998.4.30)

(71)出願人　000002185
ソニー株式会社
東京都港区港南１丁目７番１号
(74)代理人　100082131
弁理士　稲本　義雄
(72)発明者　藤波　靖
東京都港区港南１丁目７番１号　ソニー株式会社内
(72)発明者　浜田　俊也
東京都港区港南１丁目７番１号　ソニー株式会社内

Ｆターム(参考)　5D044 AB05　AB07　BC02　CC04　DE04
DE12　DE96　EF03　FG10

(54)【発明の名称】記録再生装置および方法、並びに記録媒体

(57)【要約】
【課題】装置の互換性を確保し、故障と誤認することを抑制する。
【解決手段】直前のplay itemに続いて、次のplay itemをシームレスに再生することができるとき、seamless_connection_flagを１に設定し、シームレス再生が保証できないとき、０に設定し、play itemをシームレスに再生することができるように記録位置を変更する処理を実行する。本発明は、記録再生装置に適用することができる。
【選択図】図４７



【特許請求の範囲】
【請求項１】
記録媒体に対してデータを記録または再生する記録再生装置において、
前記データを前記記録媒体に対して記録する記録手段と、
前記データを前記記録媒体に記録したとき、そのデータを、既に記録されているデータに対して連続して再生可能であるか否かを判定する判定手段とを備え、
前記記録手段は、前記判定手段により前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録を行う
ことを特徴とする記録再生装置。
【請求項２】
前記データは、少なくとも第１のデータ区間と、前記第１のデータ区間の再生時間よりも後の第２のデータ区間とを有し、
前記判定手段は、少なくとも、前記第１のデータ区間から前記第２のデータ区間を再生する間に読み出しチャネル用バッファが空になるとき、前記データの連続再生が不可能であると判定する
ことを特徴とする請求項１に記載の記録再生装置。
【請求項３】
記録媒体に対してデータを記録または再生する記録再生装置の記録再生方法において、
前記データを前記記録媒体に対して記録する記録ステップと、
前記データを前記記録媒体に記録したとき、そのデータを、既に記録されているデータに対して連続して再生可能であるか否かを判定する判定ステップとを含み、
前記記録ステップは、前記判定ステップで前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録を行う
ことを特徴とする記録再生方法。
【請求項４】
記録媒体に対してデータを記録または再生する記録再生装置に、
前記データを前記記録媒体に対して記録する記録ステップと、
前記データを前記記録媒体に記録したとき、そのデータを、既に記録されているデータに対して連続して再生可能であるか否かを判定する判定ステップとを含み、
前記記録ステップは、前記判定ステップで前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録を行う
処理を実行させることを特徴とするプログラムが記録されている記録媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、記録再生装置および方法、並びに記録媒体に関し、特に、データの連続再生に関連して、ユーザが故障と誤認するようなことを抑制するようにした記録再生装置および方法、並びに記録媒体に関する。
【背景技術】
【０００２】
ディスクには、ビデオデータやオーディオデータなどが連続して記録されるだけでなく、断続的に時間をおいて記録されることがある。また、一旦記録されたデータが消去され、他のデータが上書きされる場合がある。
【発明の開示】
【発明が解決しようとする課題】
【０００３】
このような消去あるいは上書き処理を繰り返し実行すると、連続して再生すべきデータが必ずしもディスク上の連続する位置に記録されず、ディスク上の離間した位置に記録される場合がある。このような場合に、装置の再生時のバッファの容量が充分ではないと、その記録位置によっては、データを連続的に再生することができず、再生データが一時的

に欠落するような場合がある。

【０００４】

しかしながら、不連続部分が発生するか否かは、装置のバッファの容量に影響されるので、装置間の互換性を確保することができず、最悪の場合、ユーザが、装置が故障したと誤認するおそれがあった。

【０００５】

本発明はこのような状況に鑑みてなされたものであり、装置の互換性を確保し、ユーザが装置が故障したと誤認するのを抑制するようにするものである。

【課題を解決するための手段】

【０００６】

本発明の一側面の記録再生装置は、記録媒体に対してデータを記録または再生する記録再生装置であって、前記データを前記記録媒体に対して記録する記録手段と、前記データを前記記録媒体に記録したとき、そのデータを、既に記録されているデータに対して連続して再生可能であるか否かを判定する判定手段とを備え、前記記録手段は、前記判定手段により前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録を行うことを特徴とする。

【０００７】

前記データには、少なくとも第１のデータ区間と、前記第１のデータ区間の再生時間よりも後の第２のデータ区間とを設けるようにさせることができ、前記判定手段には、少なくとも、前記第１のデータ区間から前記第２のデータ区間を再生する間に読み出しチャネル用バッファが空になるとき、前記データの連続再生が不可能であると判定させるようにすることができる。

【０００８】

本発明の一側面の記録再生方法は、記録媒体に対してデータを記録または再生する記録再生装置の記録再生方法であって、前記データを前記記録媒体に対して記録する記録ステップと、前記データを前記記録媒体に記録したとき、そのデータを、既に記録されているデータに対して連続して再生可能であるか否かを判定する判定ステップとを含み、前記記録ステップは、前記判定ステップで前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録を行うことを特徴とする。

【０００９】

本発明の一側面の記録媒体のプログラムは、記録媒体に対してデータを記録または再生する記録再生装置であって、前記データを前記記録媒体に対して記録する記録ステップと、前記データを前記記録媒体に記録したとき、そのデータを、既に記録されているデータに対して連続して再生可能であるか否かを判定する判定ステップとを含み、前記記録ステップは、前記判定ステップで前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録を行う処理を実行させることを特徴とする。

【００１０】

本発明の一側面の記録再生装置および方法、並びに記録媒体は、記録媒体に対してデータを記録または再生する記録再生装置および方法、並びに記録媒体であって、前記データが前記記録媒体に対して記録され、前記データが前記記録媒体に記録されたとき、そのデータが、既に記録されているデータに対して連続して再生可能であるか否かが判定され、前記データの連続再生が不可能であると判定された場合に、前記データのうち、一部について再記録が行われる。

【発明の効果】

【００１１】

本発明の一側面によれば、記録再生装置のバッファの容量の違いに拘らず、互換性を確保することが可能となり、ユーザが装置の故障と誤認するのを抑制することが可能となる。

【発明を実施するための最良の形態】

【００１２】

最初に本発明において情報が記録または再生される記録媒体（メディア）上のファイル配置について説明する。メディア上には、図１に示すように、次の７種類のファイルが記録される。

VOLUME.TOC
ALBUM.STR
PROGRAM_$$$.PGI
TITLE_###.VDR
CHUNKGROUP_@@@.CGIT
CHUNK_%%%%.ABST
CHUNK_%%%%.MPEG2

【００１３】

ルートディレクトリにはVOLUME.TOCおよびALBUM.STRが置かれる。また、ルートディレクトリ直下のディレクトリ"PROGRAM"には、"PROGRAM_$$$.PGI"（ここで"$$$"はプログラム番号を表す）が置かれる。同様に、ルートディレクトリ直下のディレクトリ"TITLE"には、"TITLE_###.VDR"（ここで"###"はタイトル番号を表す）が、ディレクトリ"CHUNKGROUP"には、"CHUNKGROUP_@@@.CGIT"（ここで"@@@"はチャンクグループ番号を表す）が、ディレクトリ"CHUNK"には、"CHUNK_%%%%.ABST"（ここで"%%%%"はチャンク番号を表す）が、それぞれ置かれる。

【００１４】

ルートディレクトリ直下のMPEGAVディレクトリには、更に１つ以上のサブディレクトリが作成され、その下に、"CHUNK_%%%%.MPEG2"（ここで%%%%はチャンク番号を表す）が置かれる。

【００１５】

VOLUME.TOCのファイルは、メディア上に１つ有るのが普通である。ただし、ROMとRAMのハイブリッド構造のメディア等、特殊な構造のメディアでは、複数存在することも有り得る。このファイルは、メディアの全体の性質を示すために用いられる。

【００１６】

VOLUME.TOCの構造は図２に示すようになっている。先頭にfile_type_idが置かれ、これにより該当ファイルがVOLUME.TOCであることが示される。次にvolume_information()が続き、最後にtext_block()が続く。

【００１７】

図３にvolume_information()の構成が示されている。これは、volume_attribute()、resume()、volume_rating()、write_protect()、play_protect()、recording_timer()を含んでいる。

【００１８】

volume_attribute()は、logical volumeの属性を記録する領域であり、図４にその詳細な構造が示されている。同図に示すように、この領域には、title_playback_mode_flag, program_playback_mode_flagなどが含まれている。

【００１９】

resume()は、メディアの再挿入時に、eject直前の状態を復元するための情報を記録する領域であり、その詳細な構造は、図５に示されている。

【００２０】

図３のvolume_rating()は、volume全体に対する視聴年齢制限を年齢やカテゴリに応じて実現するための情報を記録する領域であり、その詳細な構造は、図６に示されている。

【００２１】

図３のwrite_protect()は、volume内に記録されているtitle, programに対する変更や、消去操作を制限する情報を記録する領域であり、その詳細な構造は、図７に示されている。

【００２２】

図３のplay_protect()は、volume内に記録されているtitle, programに対する再生許可

、不許可の設定、あるいは、再生回数を制限する情報を記録する領域であり、その詳細な構造は、図８に示されている。

【００２３】

図３のrecording_timer()は、記録時間を制御する情報を記録する領域であり、その詳細な構造は、図９に示されている。

【００２４】

図２のVOLUME.TOCのtext_block()の詳細な構造は図１０に示されている。このtext_block()には、language_set()とtext_itemが含まれており、その詳細な構造は図１１と図１２にそれぞれ示されている。

【００２５】

図１のALBUM.STRのファイルは、メディア上に１つ有るのが普通である。ただし、ROMとRAMのハイブリッド構造のメディア等、特殊な構造のメディアでは、複数存在することも有り得る。このファイルは、複数のメディアを組み合わせて、あたかも１つのメディアであるような構成にするために使用される。

【００２６】

このALBUM.STRの構造は、図１３に示すようになっている。先頭にfile_type_idが置かれ、該当ファイルがALBUM.STRであることを示す。次にalbum()が続き、最後にtext_block()が続く。

【００２７】

album()は、複数のvolume（複数のメディア）を１つのまとまりとして扱うための情報を記録する領域であり、その詳細な構造は、図１４に示されている。

【００２８】

図１のTITLE_###.VDRのファイルは、タイトルの数だけ存在する。タイトルとは、例えばcompact discで言うところの１曲や、テレビ放送の１番組を言う。この情報の構造は図１５に示すようになっている。先頭にfile_type_idが置かれ、これにより該当ファイルがTITLE_###.VDRであることが示される。次にtitle_info()が続き、最後にtext_block()が続く。###はタイトル番号を示す文字列である。

【００２９】

title_info()は、chunkgroup上における、titleの開始点、終了点、その他titleに関する属性を記録するための領域であり、その詳細な構造は、図１６に示されている。

【００３０】

図１のPROGRAM_$$$.PGIのファイルは、プログラムの数だけ存在する。プログラムは、タイトルの一部（あるいは全部）の領域を指定した複数のカットで構成され、各カットは指定された順番で再生される。この情報の構造は図１７に示されている。先頭にfile_type_idが置かれ、該当ファイルがPROGRAM_$$$.PGIであることを示す。次にprogram()が続き、最後にtext_block()が続く。$$$はタイトル番号を示す文字列である。

【００３１】

program()は、素材に対して不可逆な編集を施すことなしに、titleの必要な部分を集めて再生するのに必要な情報を記録する領域であり、その詳細な構造は、図１８に示されている。

【００３２】

図１８のprogram()は、１つのplay_listを有している。このplay_list()の詳細は、図１９に示されている。

【００３３】

play_listには、play_item()が複数置かれている。play_item()の詳細は、図２０に示されている。

【００３４】

図１のCHUNKGROUP_@@@.CGITのファイルは、チャンクグループの数だけ存在する。チャンクグループはビットストリームを並べるためのデータ構造である。このファイルは、ユーザがVDR（ビデオディスクレコーダ）など、メディアを記録再生する装置を普通に操作

している分にはユーザに認識されない。

【００３５】
この情報の構造は図２１に示すようになっている。先頭にfile_type_idが置かれ、該当ファイルがCHUNKGROUP_@@@.CGITであることを示す。その次にchunkgroup_time_base_flagsとchunkgroup_time_base_offsetが有り、次にchunk_connection_info()、最後にtext_block()が続く。

【００３６】
chunkgroup_time_base_flagsは、chunkgroupの基準カウンタに関するflagを示し、chunkgroup_time_base_offsetは、chunkgroup内の基準時間軸の開始時刻を示す。これは、９０kHzでカウントアップするカウンタにセットする値であり、３２ビットの大きさを有する。chunk_connection_info()は、videoの切換点や、videoとaudioの同期など、特異な点の情報を記憶する領域であり、その詳細な構造は、図２２に示されている。

【００３７】
このchunk_connection_info()には、チャンクグループに属するチャンクの数だけchunk_arrangement_info()のループが置かれる。図２３にこのchunk_arrangement_info()の詳細が示されている。

【００３８】
図１のCHUNK_%%%%.ABSTのファイルは、チャンクの数だけ存在する。チャンクはストリームファイル１つに対応する情報ファイルである。この情報の構造は図２４に示すようになっている。先頭にfile_type_idが置かれ、これにより、該当ファイルがCHUNK_%%%%.ABSTであることが示される。

【００３９】
図１のCHUNK_%%%%.MPEG2のファイルは、ストリームファイルである。このファイルはMPEGのビットストリームを格納しており、この他のファイルが情報のみを記録しているのと異なっている。

【００４０】
図２５は、以上のようなファイルを有するメディアとしての光ディスクに対して情報を記録または再生する光ディスク装置の構成例を表している。この光ディスク装置では、１枚の書き換え型の光ディスク１に対して１系統の光ヘッド２が設けられており、データの読み出しと書き込みの双方にこの光ヘッド２が共用される。

【００４１】
光ヘッド２により光ディスク１から読み出されたビットストリームは、ＲＦおよび復調／変調回路３で復調された後、ECC回路４で誤り訂正が施され、スイッチ５を介して、読み出しレートとデコード処理レートとの差を吸収するための読み出しチャネル用バッファ６に送られる。読み出しチャネル用バッファ６の出力はデコード７に供給されている。読み出しチャネル用バッファ６はシステムコントローラ１３から読み書きができるように構成されている。

【００４２】
読み出しチャネル用バッファ６から出力されたビットストリームは、デコーダ７でデコードされ、そこからビデオ信号とオーディオ信号が出力される。デコーダ７から出力されたビデオ信号は合成回路８に入力され、OSD（On Screen Display）制御回路９が出力するビデオ信号と合成されたた後、出力端子Ｐ１から図示せぬディスプレイに出力され、表示される。デコーダ７から出力されたオーディオ信号は、出力端子Ｐ２から図示せぬスピーカに送られて再生される。

【００４３】
他方、入力端子Ｐ３から入力されたビデオ信号、および入力端子Ｐ４から入力されたオーディオ信号は、エンコーダ１０でエンコードされた後、エンコード処理レートと書き込みレートとの差を吸収するための書き込みチャネル用バッファ１１に送られる。この書き込みチャネル用バッファ１１もシステムコントローラ１３から読み書きができるように構

成されている。

【００４４】

書き込みチャネル用バッファ１１に蓄積されたデータは、書き込みチャネル用バッファ１１から読み出され、スイッチ５を介してECC回路４に入力されて誤り訂正符号が付加された後、ＲＦおよび復調／変調回路３で変調される。ＲＦおよび復調／変調回路３より出力された信号（ＲＦ信号）は、光ヘッド２により光ディスク１に書き込まれる。

【００４５】

アドレス検出回路１２は、光ディスク１の記録または再生するトラックのアドレス情報を検出する。システムコントローラ１３は、この光ディスク装置の各部の動作を制御するものであり、各種の制御を行うCPU２１、CPU２１が実行すべき処理プログラム等を格納したROM２２、処理過程で生じたデータ等を一時記憶するためのRAM２３、および光ディスク１に対して記録または再生する各種の情報ファイルを記憶するRAM２４を有している。CPU２１は、アドレス検出回路１２の検出結果に基づいて、光ヘッド２の位置を微調整する。CPU２１はまた、スイッチ５の切り替え制御を行う。各種のスイッチ、ボタンなどから構成される入力部１４は、各種の指令を入力するとき、ユーザにより操作される。

【００４６】

次に、基本的な情報ファイルの読み込み動作について説明する。例えば、"VOLUME.TOC"情報ファイルの読み込みを行うとき、ステムコントローラ１３のCPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、"VOLUME.TOC"が記録されている光ディスク１上の物理アドレスと、その長さを確定する。続いて、CPU２１は、この"VOLUME.TOC"のアドレス情報に基づき、光ヘッド２を読み出し位置に移動させる。そしてCPU２１は、光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を読み出しモードに設定するとともに、スイッチ５を読み出しチャネル用バッファ６側に切り替え、さらに光ヘッド２の位置を微調整した後、光ヘッド２による読み出しを開始させる。これにより"VOLUME.TOC"の内容が光ヘッド２により読み出され、ＲＦおよび復調／変調回路３により復調され、さらにECC回路４により誤り訂正が行われた後、読み出しチャネル用バッファ６に蓄積される。

【００４７】

読み出しチャネル用バッファ６に蓄積されたデータ量が、"VOLUME.TOC"の大きさと等しいか、あるいはより大きくなった時点で、CPU２１は読み出しを停止させる。その後、CPU２１は、読み出しチャネル用バッファ６から該当データを読み出し、RAM２４に記憶させる。

【００４８】

次に、基本的な情報ファイル書込み動作について、"VOLUME.TOC"情報ファイルを書き込む場合を例として説明する。CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク１）中に、これから書こうとしている"VOLUME.TOC"と等しいか、より大きい大きさを持つ空き領域を探し、そのアドレスを確定する。

【００４９】

次に、CPU２１は、RAM２４に用意されている、新たに書き込むべき"VOLUME.TOC"を、書き込みチャネル用バッファ１１に転送する。続いて、CPU２１は、空き領域のアドレス情報に基づき、光ヘッド２を書き込み位置に移動させる。そしてCPU２１は、光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を書き込みモードに設定するとともに、スイッチ５を書き込みチャネル用バッファ１１側に切り替え、光ヘッド２の位置を微調整した後、光ヘッド２による書き込みを開始させる。

【００５０】

これにより新たに用意した"VOLUME.TOC"の内容が、書き込みチャネル用バッファ１１から読み出され、スイッチ５を介してECC回路４に入力され、誤り訂正符号が付加された後、ＲＦおよび復調／変調回路３により変調される。ＲＦおよび復調／変調回路３より出力された信号は、光ヘッド２により光ディスク１に記録される。書き込みチャネル用バッフ

ファ１１から読み出され、光ディスク１に記録されたデータ量が、"VOLUME.TOC"の大きさと等しくなった時点で、CPU２１は書き込み動作を停止させる。

【００５１】

最後に、CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク１）中の"VOLUME.TOC"を指し示すポインタを、新しく書込んだ位置を指し示すように書き換える。

【００５２】

次に、基本的なストリーム再生動作について、図１のCHUNK_0001.MPEG2というストリームを再生する場合を例として説明する。CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、"CHUNK_0001.MPEG2"が記録されている光ディスク１上の物理アドレスと、その長さを確定する。続いて、CPU２１は、この"CHUNK_0001.MPEG2"のアドレス情報に基づき、光ヘッド２を読み出し位置に移動させる。そして光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を読み出しモードに設定するとともに、スイッチ５を読み出しチャネル用バッファ６側に切り替え、光ヘッド２の位置を微調整した後、光ヘッド２による読み出しを開始させる。

【００５３】

光ヘッド２により読み出された"CHUNK_0001.MPEG2"の内容が、ＲＦおよび復調／変調回路３、ECC回路４、並びにスイッチ５を介して読み出しチャネル用バッファ６に蓄積される。読み出しチャネル用バッファ６に蓄積されたデータは、デコーダ７に出力され、デコード処理が施されて、ビデオ信号とオーディオ信号がそれぞれ出力される。オーディオ信号は出力端子Ｐ２から出力され、ビデオ信号は、合成回路８を介して出力端子Ｐ１から出力される。

【００５４】

光ディスク１から読みだされ、デコード、表示されたデータ量が、"CHUNK_0001.MPEG2"の大きさと等しくなった時点で、あるいは、入力部１４から読み出し動作の停止が指定された時点で、CPU２１は、読み出しおよびデコード処理を停止させる。

【００５５】

次に、基本的なストリーム記録動作を、"CHUNK_0001.MPEG2"情報ファイルを書き込む場合を例として説明する。CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク１）中にこれから書こうとしている"CHUNK_0001.MPEG2"と等しいか、それより大きい大きさを持つ空き領域を探し、そのアドレスを確定する。

【００５６】

入力端子Ｐ３から入力されたビデオ信号、および入力端子Ｐ４から入力されたオーディオ信号は、エンコーダ１０によりエンコードされた後、書き込みチャネル用バッファ１１に蓄積される。続いて、CPU２１は、空き領域のアドレス情報に基づき、光ヘッド２を書き込み位置に移動させる。そしてCPU２１は、光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を書き込みモードに設定するとともに、スイッチ５を書き込みチャネル用バッファ１１側に切り替え、光ヘッド２の位置を微調整した後、光ヘッド２による書き込みを開始させる。これにより新たに用意した"CHUNK_0001.MPEG2"の内容が、書き込みチャネル用バッファ１１から読み出され、スイッチ５、ECC回路４、ＲＦおよび復調／変調回路３を介して光ヘッド２に入力され、光ディスク１に記録される。

【００５７】

書き込みチャネル用バッファ１１から読み出され、光ディスク１に記録されたデータ量が、予め設定した値と等しくなったとき、あるいは入力部１４から書き込み動作の停止が指定されたとき、CPU２１は書き込み動作を停止させる。最後に、CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム（光ディスク１）中の"CHUNK_0001.MPEG2"を指し示すポインタを、新しく書込んだ位置を指し示すように書き換える。

【００５８】

いま、光ディスク１に、図２６に示すような情報ファイルとストリームファイルが記録されているものとする。この例では、"PROGRAM_001.PGI"という名前の１つのプログラムのファイルが含まれている。また、この光ディスク１には、"TITLE_001.VDR"、"TITLE_002.VDR"、および"TITLE_003.VDR"という名前の３つのタイトルのファイルが含まれている。

【００５９】

さらに、この光ディスク１には、"CHUNKGROUP_001.CGIT"と"CHUNKGROUP_002.CGIT"という２つのチャンクグループのファイルが含まれている。また、この光ディスク１には、"CHUNK_0001.MPEG2"、"CHUNK_0011.MPEG2"、および"CHUNK_0012.MPEG2"という名前の３つのストリームのファイルが含まれているとともに、それぞれに対応する情報として、"CHUNK_0001.ABST"、"CHUNK_0011.ABST"、および"CHUNK_0012.ABST"の３つの情報ファイルが置かれている。

【００６０】

図２６に示した情報ファイルとストリームファイルを有する光ディスク１の論理構造は、図２７に示すようになる。この例では、チャンク情報ファイル"CHUNK_0001.ABST"は、ストリームファイル"CHUNK_0001.MPEG2"を、またチャンク情報ファイル"CHUNK_0011.ABST"は、ストリームファイル"CHUNK_0011.MPEG2"を、さらに、チャンク情報ファイル"CHUNK_0012.ABST"は、ストリームファイル"CHUNK_0012.MPEG2"を、それぞれ指定している。具体的には、図２４のCHUNK_%%%%.ABST中の、chunk_file_idというフィールドで、ストリームのファイルIDが指定される。

【００６１】

さらに、この例では、チャンクグループ情報ファイル"CHUNKGROUP_001.CGIT"は、チャンク情報ファイル"CHUNK_0001.ABST"を、またチャンクグループ情報ファイル"CHUNKGROUP_002.CGIT"は、チャンク情報ファイル"CHUNK_0011.ABST"と"CHUNK_0012.ABST"を、それぞれ指定している。具体的には、図２３のchunk_arrangement_info()の中のchunk_info_file_idというフィールドでチャンク情報のファイルIDが指定される。このchunk_arrangement_info()はチャンクグループ情報ファイルの中にあり、該当チャンクグループに属するチャンクの数だけ存在するデータ構造となっている（図２３のchunk_arrangement_info()は、図２２のchunk_connection_info()に記述されており、このchunk_connection_info()は、図２１のCHUNKGROUP_###.CGITに記述されている）。

【００６２】

CHUNKGROUP_001には、chunk_arrangement_info()が１つだけあり、その中のchunk_info_file_idがCHUNK_0001を指定している。CHUNKGROUP_002には、chunk_arrangement_info()が２つあり、その中で、それぞれCHUNK_0011とCHUNK_0012が指定されている。このような場合のため、チャンクグループは、複数のチャンクの再生順序等を指定できるようになっている。

【００６３】

具体的には、まず、図２１のCHUNKGROUP_###.CGIT中のchunkgroup_time_base_offsetにより、該当チャンクグループでの時計の初期値が定められる。次に各チャンクを登録する際に、図２３のchunk_arrangement_info()のpresentation_start_cg_countとpresentation_end_cg_time_countが指定される。

【００６４】

例えば、図２８に示すように、CHUNK_0011の長さ（時間）をA、CHUNK_0012の長さ（時間）をBとする。CHUNK_0011のpresentation_start_cg_countがchunkgroup_time_base_offsetに等しく、presentation_end_cg_countが"chunk_group_time_base_offset+A"に等しい。またCHUNK_0012のpresentation_start_cg_countがchunkgroup_time_base_offset+Aに等しく、presentation_end_cg_countが"chunk_group_time_base_offset+A+B"に等しい。このように設定すると、CHUNKGROUP_002は、CHUNK_0011とCHUNK_0012を連続的に再生させたものとして定義される。

【００６５】

なお、CHUNK_0011とCHUNK_0012の再生時刻に重なりがある場合には、時刻をそのようにずらすことで指定ができる。また、図２３のchunk_arrangement_info()中のtransition_info()に記述を行うことで、２つのストリーム間の遷移において、特殊効果（フェードイン、フェードアウト、ワイプ等）を指定できるようになっている。

【００６６】

図２６（図２７）の例では、タイトル情報ファイル"TITLE_001.VDR"と"TITLE_002.VDR"は、チャンクグループ情報ファイル"CHUNKGROUP_001.CGIT"を、またタイトル情報ファイル"TITLE_003.VDR"はチャンクグループ情報ファイル"CHUNKGROUP_002.CGIT"を、それぞれ指定している。具体的には、図１６のtitle_info()中において、cgit_file_idというフィールドで、チャンクグループのファイルIDを指定し、さらにtitle_start_chunk_group_time_stampとtitle_end_chunk_group_time_stampというフィールドで、チャンクグループ内で該当タイトルが定義される時間的な範囲を指定している。

【００６７】

例えば、図２７の例では、CHUNKGROUP_001の前半をTITLE_001が、後半をTITLE_002が、それぞれ指し示している。なお、この分割はユーザからの要求により行われたものであり、その位置はユーザにとって任意であり、予め決めておくことはできない。ここでTITLE_001とTITLE_002による分割の位置を、CHUNKGROUP_001の先頭からＡだけ離れた位置に設定したとする。

【００６８】

TITLE_001はチャンクグループとしてCHUNKGROUP_001を指定し、タイトルの開始時刻として、CHUNKGROUP_001の開始時刻を指定し、タイトルの終了時刻として、ユーザから指定された点の時刻を指定する。

【００６９】

つまりTITLE_001のtitle_start_chunk_group_time_stampとして、CHUNKGROUP_001のchunkgroup_time_base_offset（先頭の位置）が設定され、TITLE_001のtitle_end_chunk_group_time_stampとして、CHUNKGROUP_001のchunkgroup_time_base_offsetにＡの長さを加えたものが設定される。

【００７０】

また、TITLE_002はチャンクグループとしてCHUNKGROUP_001を指定し、タイトルの開始時刻として、ユーザから指定された点の時刻を指定し、タイトルの終了時刻として、CHUNKGROUP_001の終了時刻を指定する。

【００７１】

つまりTITLE_002のtitle_start_chunk_group_time_stampとして、CHUNKGROUP_001のchunkgroup_time_base_offset（先頭の位置）にＡの長さを加えたものが設定され、TITLE_002のtitle_end_chunk_group_time_stampとして、CHUNKGROUP_001のchunkgroup_time_base_offsetにCHUNKGROUP_001の長さを加えたものが設定される。

【００７２】

さらに、TITLE_003はチャンクグループとしてCHUNKGROUP_002を指定し、タイトルの開始時刻としてCHUNKGROUP_002の開始時刻を指定し、タイトルの終了時刻としてCHUNKGROUP_002の終了時刻を指定する。

【００７３】

つまりTITLE_003のtitle_start_chunk_group_time_stampとして、CHUNKGROUP_002のchunkgroup_time_base_offsetが設定され、TITLE_003のtitle_end_chunk_group_time_stampとして、CHUNKGROUP_002のchunkgroup_time_base_offsetにCHUNKGROUP_002の長さを加えたものが設定される。

【００７４】

さらに、この例では、プログラム情報ファイル"PROGRAM_001.PGI"は、TITLE_001の一部とTITLE_003の一部を、この順番で再生するように指定している。具体的には、図２０のplay_item()中のtitle_numberによりタイトルを指定し、各タイトルで定義される時刻で開始点と終了点を定義することで、１つのカットが抜き出される。このようなカットを複数

個集めて、プログラムが構成される。

【００７５】

次に、光ディスク１に、新たな情報を追記録（アペンド記録）する場合の動作について説明する。この記録は、具体的には、例えば、タイマ録画により、あるいはユーザが入力部１４を操作して、光ディスク装置に対してリアルタイムに録画を指令することにより行われる。後者の場合、録画ボタンが押されたようなときは、録画終了時刻を予測することはできないが、ワンタッチ録画機能（操作後、一定時間だけ録画が行われる機能）のボタンが押されたときは、終了時刻を予測することができる。

【００７６】

ここではタイマ録画を例にとって説明する。この場合、光ディスク装置のユーザは事前に、録画開始時刻、録画終了時刻、ビットストリームのビットレート、録画を行うチャネル等を指定してあるものとする。また、録画の予約を行った時点で、ビットレートと録画時間に見合う空き容量が光ディスク１に残されていることが、予め確認されているものとする。

【００７７】

記録予約時と予約された記録の実行時との間に、光ディスク１に対して更なる記録が行われたような場合、今回予約された番組を、指定されたビットレートで記録する分の容量を確保することができなくなる場合がある。このような場合、CPU２１は、ビットレートを、指定された値より下げて、予約された時間分の情報を記録するようにするか、または、ビットレートはそのままにして、記録可能な時間だけ記録するようにする。CPU２１は、このとき、更なる記録が行われ、予約した記録に不具合が出た時点でユーザにその旨を伝えるメッセージを発することは言うまでもない。

【００７８】

さて、予約された録画の開始時刻が近づくと、CPU２１は内蔵するタイマやクロックを使用して、モードを、スリープモードから動作モードに自動的に復帰させる。そしてCPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、予約された番組が記録できるだけの領域を光ディスク１上に確保する。つまり、予約録画の終了時刻から開始時刻を減算した結果（録画時間）にビットレートを乗じた数値が、予約された番組を記録するのに必要な領域の大きさであり、CPU２１はこの大きさの領域をまず確保する。その他、この記録に際して、ストリームファイル以外に情報ファイルを記録する必要がある場合、例えば新たなタイトルとして登録するためにタイトル情報ファイル等が必要である場合には、それらの情報ファイルが記録できるだけの容量を光ディスク１に確保しておく必要がある。必要な分の領域を確保することができない場合には、上述したような方法（ビットレートの変更、録画可能な時間内だけの録画などの方法）で対応が取られることになる。

【００７９】

なおこのとき、新しいタイトルの記録なので、ユーザは、新たなストリームディレクトリの新たなストリームファイルとして新しいストリームファイルのファイル名を付ける。ここでは、これを、\MPEGAV\STREMS_003\CHUNK_0031とする。つまり、図２９に示すように、ルートディレクトリの下のMPEGAVディレクトリの下のSTREAM_003ディレクトリの下のCHUNK_0031.MPEG2という名前のファイルとする。

【００８０】

CPU２１は、各部に対して記録モードの実行を命令する。例えば、図示せぬチューナから入力端子Ｐ３に入力されたビデオ信号、および入力端子Ｐ４に入力されたオーディオ信号は、エンコーダ１０によりエンコードされた後、書き込みチャネル用バッファ１１に蓄積される。続いて、CPU２１は、先程確保した領域のアドレス情報に基づき、光ヘッド２を書き込み位置に移動させる。そしてCPU２１は、光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を書き込みモードに設定するとともに、スイッチ５を書き込みチャネル用バッファ１１側に切り替え、光ヘッド２の位置を微調整した後、光ヘッド２による書き込みを開始させる。これにより新たに用意した"CHUNK_0031.MPEG2"の内容が、書き

込みチャネル用バッファ１１から読み出され、スイッチ５、ECC回路４、ＲＦおよび復調／変調回路３、並びに光ヘッド２を介して、光ディスク１に記録される。

【００８１】

以上の書き込み動作を続けて、以下のいずれかの条件が発生した時点で、CPU２１は、書き込み動作を停止させる。

１）予約された記録の終了時刻になったとき

２）容量不足、その他の原因により光ディスク１に記録ができなくなったとき

３）録画動作の停止が指令されたとき

【００８２】

次に、CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、ファイルシステム中の"CHUNK_0031.MPEG2"を指し示すポインタを新しく書込んだ位置を指し示す値に書き換える。また、CPU２１は、チャンク情報、チャンクグループ情報、タイトル情報のそれぞれのファイルを用意し、しかるべき名前をつけて記録する。なお、記録時あるいは予約時に、光ディスク１上に、これらのファイルを記録することができるだけの空き容量を確保しておく必要がある。

【００８３】

このようにして、例えば図３０に示すように、新たな情報ファイルが作成される。同図において、ファイル名の右肩にアスタリスク(*)をつけたものが、今回新たに作成されたファイルである。

【００８４】

図３１は、新たにでき上がった情報ファイルの関係を示したものである。TITLE_004はCHUNKGROUP_003を指定し、CHUNKGROUP_003はCHUNK_0031を指定し、CHUNK_0031はSTREAM_0031を指定している。

【００８５】

すなわち、新たなストリームはTITLE_004として、情報ファイルに登録されている。ユーザは光ディスク装置のタイトルを確認する機能により、TITLE_004の属性等を知ることができ、また、TITLE_004を再生することができる。

【００８６】

次に、図２６（図２７）に例示するような光ディスク１上に、上書き記録する場合の動作について説明する。上書き記録とは、ビデオテープに信号を記録する場合と同様に、それまでに記録されている番組の上に（その番組を消去して）新たな番組を記録していく動作のことを言う。

【００８７】

上書き記録では、上書き記録を開始する位置が重要である。例えばユーザからTITLE_001の先頭から上書き記録を開始することが指定されたとする。この時上書き記録は、TITLE_001、TITLE_002、TITLE_003をそれぞれ順に書き換えながら行われる。TITLE_003の最後まで書き換えてもまだ記録動作が終了しない場合には、光ディスク１上の空き領域に新たな領域を確保して記録が続行される。例えばTITLE_002が記録開始位置とされた場合には、TITLE_001は記録開始位置より前に位置するので、今回の記録動作により書き換えられることはない。

【００８８】

いま、TITLE_003の先頭からタイマ録画により上書きするものとする。この場合、光ディスク装置のユーザは事前に、録画開始時刻、終了時刻、ビットストリームのビットレート、録画を行うチャネル等を指定しているものとする。また、上書き記録では重要な記録開始位置がTITLE_003の先頭と指定されたものとする。さらにこの場合においても、録画の予約を行った時点で、ビットレートと録画時間に見合う容量が光ディスク１上に存在することが、予め確認されているものとする。上書き記録の場合には、指定された位置から上書き可能な（複数の）タイトルの総容量と、光ディスク１の空き容量の和が記録可能容量となる。つまり、今回の場合には、TITLE_003が管理するストリームSTREAM_0011とSTREAM_0012の総容量と、光ディスク１上の空き容量の和が記録可能な容量となる。

【００８９】
上書き記録では、記録可能な容量分に対して、どのような順番で実際の記録を行なっていくかという選択肢がいくつかある。まず、最初に考えられるのがタイトルで指定されているストリームの順番に記録していく方法である。つまり、今回の場合には、まずSTREAM_0011の先頭から記録を開始し、STREAM_0011の終わりまで記録したら、STREAM_0012の先頭から記録を続行し、STREAM_0012の終わりまで記録したら、今度は空き領域に記録を行なう方法である。もう１つの方法は、まず、空き領域に記録を行い、空き領域が無くなった時点で、現存するストリーム上に記録していく方法である。
【００９０】
前者の方法は、ビデオテープのエミュレーションという意味で優れている。つまり、ビデオテープと同様の動作であるという意味で、ユーザから理解され易いという特徴を有する。後者の方法は、既に記録されているストリームの消去が後回しにされるため、記録されているものの保護という点で優れていると言う特徴を有する。
【００９１】
なお、記録予約時と予約された記録の実行時との間に、光ディスク１に対して更なる記録が行われた場合に、今回予約された番組を、指定されたビットレートで記録する分の容量を確保することができない場合がある。このような場合、上述した場合と同様に、予約実行時に、ビットレートが自動的に下げられ、予約された時間分だけすべて記録されるか、または、ビットレートはそのままにして、記録可能な時間だけ記録が行われる。
【００９２】
予約された録画の開始時刻が近づくと、光ディスク装置はスリープモードから動作モードに復帰する。CPU２１は、光ディスク１上の空き容量をすべて確保する。もちろん、この時点で空き容量を確保せず、必要になった時点で確保するという方法もあるが、ここでは説明のために、記録開始以前に必要な領域を確保するものとする。
【００９３】
なお、タイマ録画等で、開始時刻、終了時刻、ビットレートが指定されているため、必要な領域の大きさが予め判っている場合には、必要な分だけ（あるいは幾分かのマージンを加えた分だけ）容量を確保するようにしてもよい。この記録に際して情報ファイルを記録する必要がある場合、例えば新たなタイトルとして登録するためにタイトル情報ファイル等が必要である場合、それらの情報ファイルも記録することができるだけの容量を残しておく必要がある。
【００９４】
ここでは、新たなストリームディレクトリの新たなストリームファイルとして新しいストリームファイルのファイル名をつけるものとする。つまり、ここでは、ファイル名を、\MPEGAV\STREMS_002\CHUNK_0031とする。すなわち、図３２に示すように、ルートディレクトリの下のMPEGAVディレクトリの下のSTREAM_002ディレクトリの下のCHUNK_0031.MPEG2という名前のファイルが作成される。
【００９５】
入力端子Ｐ３に入力されたビデオ信号、および入力端子Ｐ４に入力されたオーディオ信号は、エンコーダ１０によりエンコードされた後、書き込みチャネル用バッファ１１に蓄積される。続いて、CPU２１は、先程確保した領域のアドレス情報に基づき、光ヘッド２を書き込み位置に移動させる。そしてCPU２１は、光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を書き込みモードに設定するとともに、スイッチ５を書き込みチャネル用バッファ１１側に切り替え、光ヘッド２の位置を微調整した後、光ヘッド２による書き込みを開始させる。これにより新たに用意した"CHUNK_0031.MPEG2"の内容が、書き込みチャネル用バッファ１１から読み出され、スイッチ５、ECC回路４、ＲＦおよび復調／変調回路３、並びに光ヘッド２を介して、光ディスク１に記録される。
【００９６】
この時、まずはストリームファイル"CHUNK_0011.MPEG2"が書き換えられる。そして"CHUNK_0011.MPEG2"の最後まで記録が行われたら、次に、"CHUNK_0012.MPEG2"へ記録が進めら

れ、さらに、"CHUNK_0031.MPEG2"へと記録が進められる。

【００９７】

以上の動作を続けて、上述した場合と同様に、３つの条件のいずれかが発生した時点で、CPU２１は、書き込み動作を停止させる。

【００９８】

次に、CPU２１は、予めその処理プログラムに組み込んであったファイルシステム操作命令を使用し、ストリームファイル、チャンク情報、チャンクグループ情報、タイトル情報を更新する。

【００９９】

ところで、書き込みが終了したタイミングによって、ファイルの構成が変化する。例えば、CHUNK_0011.MPEG2とCHUNK_0012.MPEG2の２つのストリームの上書きを終了した後、さらにCHUNK_0031.MPEG2に記録が行われた場合、光ディスク１のファイルの構成は、図３３に示すようになる。ファイル名の右肩にアスタリスク(*)をつけたものが今回新たに作成されたファイルである。

【０１００】

図３４は、このようにして新たにでき上がったファイル（図３３のファイル）の関係を示したものである。図３１と比較して明らかなように、TITLE_003が指定しているCHUNKGROUP_002に含まれるCHUNKとしてCHUNK_0031が増えており、CHUNK_0031はSTREAM_0031を指定している。

【０１０１】

一方、既存ストリームの上書きの途中で上書き記録が終了した場合、例えば、CHUNK_0011の記録の途中で上書き記録が終了した場合、上書きのために確保したCHUNK_0031のストリームは上書きされなかったので開放される。この場合、特殊なタイトルの処理が行われる。すなわち、TITLE_003の先頭から上書き記録を開始し、その途中で記録が終了した場合には、そこでタイトルが分割される。つまり、図３５に示すように、上書き記録開始位置から終了位置までが新たなTITLE_003とされ、それ以降の（元々のTITLE_003の残り部分）はTITLE_004とされる。

【０１０２】

次に、タイトル再生の動作について説明する。いま、図２６に示すようなファイルを有する光ディスク１を光ディスク装置に挿入し、タイトル再生するものとする。まず、光ディスク１が挿入されると、CPU２１は情報ファイルを光ディスク１から読み込んで、RAM２４に記憶させる。この動作は上述した、基本的な情報ファイルの読み込み動作を繰り返すことで行われる。

【０１０３】

CPU２１は、まず、VOLUME.TOCとALBUM.STRを読み出す。次にCPU２１は、ディレクトリ"TITLE"以下に、".VDR"の拡張子を持つファイルがいくつ有るかを調べる。この拡張子を持つファイルは、タイトルの情報を持つファイルであり、そのファイルの数はつまりタイトルの数となる。図２６の例ではタイトル数は３となる。次にCPU２１は３つのタイトル情報ファイルを読み込み、RAM２４に記憶させる。

【０１０４】

CPU２１は、OSD制御回路９を制御して、光ディスク１上に記録されているタイトルの情報を示す文字情報を発生させ、合成回路８によりビデオ信号と合成させ、出力端子Ｐ１からディスプレイに出力させ、表示させる。いまの場合、タイトルが３つあること、そして３つのタイトルそれぞれの長さや属性（名前、記録された日時など）が表示される。

【０１０５】

ここで、ユーザが、例えばTITLE_002の再生を指定したとする。TITLE_002の情報ファイルには（図１６のtitle_info()中のcgit_file_idには）、CHUNKGROUP_001を指定するファイルIDが記録されており、CPU２１はこれを記憶するとともに、CHUNKGROUP_001をRAM２４に格納させる。

【０１０６】

次に、CPU２１は、TITLE_002の開始時刻と終了時刻（図１６のtitle_info()中のtitle_start_chunk_group_time_stampとtitle_end_chunk_group_time_stamp）が、どのCHUNKに対応するかを調べる。これは、CHUNKGROUPの情報の中から、それぞれのCHUNKが登録されている情報（図２３のchunk_arrangement_info()中のpresentation_start_cg_time_countとpresentation_end_cg_time_count）を比較することで行なわれる。いまの場合、図２７に示すように、TITLE_002の開始時刻は、CHUNK_0001の途中に入っていることがわかる。つまり、TITLE_002を先頭から再生するには、ストリームファイル"CHUNK_0001.MPEG2"の途中から再生を開始すれば良いと言うことがわかる。

【０１０７】

次に、CPU２１は、TITLE_002の先頭がストリーム中のどこにあたるかを調べる。すなわち、TITLE_002の開始時刻が、ストリーム中のオフセット時刻（タイムスタンプ）としていくつにあたるのかが計算され、次にCHUNKファイル中の特徴点情報を使用して、開始時刻直前にあたる再生開始点が特定される。これにより、再生開始点のファイル先頭からのオフセット距離が確定できたことになる。

【０１０８】

次に、CPU２１は、予めその処理プログラムに組み込んであるファイルシステム操作命令を使用し、"CHUNK_0001.MPEG2"が記録されている光ディスク１上の物理アドレスと、その長さを確定する。更に、このアドレスに、先程求めた再生開始点のオフセットアドレスが加えられて、TITLE_002の再生開始点のアドレスが最終的に確定される。

【０１０９】

続いて、CPU２１は、この"CHUNK_0001.MPEG2"のアドレス情報に基づき、光ヘッド２を読み出し位置に移動させる。そしてCPU２１は、光ヘッド２、ＲＦおよび復調／変調回路３、並びにECC回路４を読み出しモードに設定するとともに、スイッチ５を読み出しチャネル用バッファ６側に切り替え、光ヘッド２の位置を微調整した後、光ヘッド２による読み出しを開始させる。これにより"CHUNK_0001.MPEG2"の内容が読み出しチャネル用バッファ６に蓄積される。

【０１１０】

読み出しチャネル用バッファ６に蓄積されたデータは、デコーダ７に出力され、デコード処理が施されて、ビデオ信号とオーディオ信号が出力される。光ディスク１から読みだされ、デコードされ、表示されたデータ量が、"CHUNK_0001.MPEG2"の大きさと等しくなった時点で、CPU２１は、TITLE_003の再生に移行する。このTITLE_003の再生動作は、TITLE_002の再生動作と同様の動作である。

【０１１１】

登録されているタイトルの再生が終了したとき、あるいは読み出し動作の停止が指示されたとき、読み出し、デコード処理が停止される。

【０１１２】

なお、光ディスク装置に、光ディスク１として、新しいディスクが挿入された場合、あるいは、異なるフォーマットのディスクが挿入された場合、CPU２１は、ディスクが挿入されたとき、VOLUME.TOCとALBUM.STRを読み出そうとするが、これらのディスクには、このようなファイルが存在しないことになる。このような場合、即ち、VOLUME.TOCとALBUM.STRを読み出すことができない場合、CPU２１はメッセージを出力し、ユーザに指示を求める。ユーザは、CPU２１に指示し、光ディスク１をイジェクトさせるか（例えば、異なるフォーマットのディスクである場合）、初期化させるか（例えば、同一フォーマットの新しいディスクである場合）、または何らかの方法によりデータを復旧させる（例えば、同一フォーマットのディスクであるが、データが破壊されている場合）。

【０１１３】

次に、titleについて、さらに説明する。図１５に示すように、TITLE_###.VDRは、titleの情報を格納するためのfileである。１つのtitleに関する情報は、１つのtitle_info()に記録される。TITLE_###.VDR内に存在するtitle_info()の数は１個である。従って、volume内にはtitleの数だけTITLE_###.VDRが存在する。

【０１１４】
title numberは、図１６のtitle_info()の中で定義せず、file名またはfile idで決定される。従って、TITLE_###.VDRのうちの正の整数###がtitle numberを表す。titleは構造というよりも、chunkgroupにつけられた、開始点を表すタイトルインデックスから、次のタイトルの先頭を表すタイトルインデックスまでの範囲、またはchunkgroupの終了点までの範囲の部分である。
【０１１５】
図１５のTITLE_###.VDRのfile_type_idは、図３６に示すように、title_info()が記録されたfileであることを示すidであり、長さ１６の文字列で表される。text_block()は、さまざまなtextを格納するための領域であり、そのtext_block()で使用が許されているtext itemだけが記述される。
【０１１６】
title_info()は、図１６に示すように、chunkgroup上における、titleの開始点と終了点、その他のtitleに関する属性が書かれる領域である。また、title_info()は、タイトル番号順に再生したとき、タイトル間でシームレス再生が保証できるか否かを示すflagを持つことができる。このflagにより、光ディスク装置でタイトル間をシームレス再生できるか否かが事前に把握でき、また併合時に配置を変える必要があるかどうかがわかる。
【０１１７】
タイトルの境界はfileの境界でも有り得るため、シームレス再生は保証されないことがある。ただし、光ディスク装置の機能として、再配置等を行うことにより、一般的にシームレス再生が行われるような状態にすることは可能である。
【０１１８】
図１６のtitle_info()中のtitle_info_lengthは、title_info()の長さをbyte単位で表したものである。flags_for_titleには、対応するtitleの書き込み属性、書き込み（変更許可）、再生回数制限、ratingのlevel等が記録される。cgit_file_idには、対応するtitleのbaseであるchunkgroupのinformation file（CHUNKGROUP_###.CGIT）のfile_idが記録される。
【０１１９】
title_start_chunk_group_time_stampには、chunkgroupで定義されたlocalな時間軸上における、そのtitleの再生開始点の時刻が記録される。そのtitleのtitleindexが指しているピクチャの表示時間がこの値となる。title_end_chunk_group_stampには、chunkgroupで定義されたlocalな時間軸上における、そのtitleの再生終了点の時刻が記録される。この値は、chunkgroupの再生終了点か、または時間軸上において直後に位置するtitleの開始点を表すtitleindexが示す値と
同一となる。
【０１２０】
title_playback_time()には、そのtitleの再生時間（タイムコード値、またはframe若しくはfield枚数）が記録される。number_of_marksには、そのtitle内に設定されているすべてのmarkの総数（titleindexを除く）が記録される。mark_typeには、図３７に示すように、title内の任意の位置につけられるmarkの種類が記録される。markは、title内のrandom access pointとしても利用される。mark_chunk_group_time_stampには、chunkgroupの時間軸上において、そのmarkが設定されている個所のtime stampが、値の小さなものから順に記録される。time stampが同じ値のmarkが複数存在してはならない。titleの開始点および終了点と同じtime stampを有するindexは存在してもよい。stuffing_bytesには、stuffingするbytesが記録され、その長さは８×n bit(n>=0)となる。
【０１２１】
次に、図２１乃至図２４に示したchunkgroupとchunkについてさらに説明する。CHUNKGROUP_###.CGITは、titleの時間軸の定義、およびchunkの構成、titleに含まれる不連続点の処理を記述したファイルである。
【０１２２】

titleは各種のbitstreamで構成されており、videoが無いstreamである場合や、DV（ディジタルビデオ）のbitstreamである場合もある。DVフォーマットでは、フレーム単位で時間軸が規定されており、MPEG2 videoのSTC（System Time Clock）を基準にしていると、フォーマットが異なるので、このDVのbitstreamを管理することができない。

【０１２３】

そこで、title内でlocalな時間軸を設定するものとする。この時間軸はtitleを構成するstreamに依存しない。titleの境界はchunkの境界とは無関係に設定される。そのため、localな時間軸は、chunk（bitstreamに１対１で対応される）毎や、title毎に設定するよりも、複数の（任意の数の）titleが含まれるchunkの集合体に対して設定するのが適切である。このchunkの集合体がchunkgroupである。

【０１２４】

chunkgroupでは、単一の時間軸を定義し、その上にchunkを貼り付けていくことでchunkの表示時刻を定めている。つまり、chunkgroupには、bitstream fileの内容（byte列）を時間軸上に展開した状態でchunkが並ぶ。１つのbitstream fileに含まれるすべてのchunkを時間軸上に並べたものをpathと呼ぶ。chunkgroupには、複数のpathを並べることができる。pathのうち、chunkgroupの再生開始時刻と終了時刻を規定するpathはmain pathと称され、その他のpathは、sub pathと称される。sub pathは、主に、後から追加記録されたaudioのchunk等を表す。

【０１２５】

chunkの接続点はtitleの境界とは必ずしも一致しないので、titleの属性ではない。しかし、chunk間の関係を各chunkの属性の含めると、階層的に矛盾が発生する場合がある。このような不連続点情報は、chunkとtitleの中間に位置するものであり、chunkgroupの階層に置くのが適切であると考えられる。

【０１２６】

以上をまとめると、chunkgroupが持つ情報は、chunkの時間軸上への配置の仕方、chunkの再生順序、chunkの終わりと次に再生するchunkの始まりとの接続点で発生する不連続点などである。

【０１２７】

図２１のCHUNKGROUP_###.CGITのfile_type_idは、図３８に示すように、そのファイルがCHUNKGROUP_###.CGITであることを表す識別子であり、ISO 646に従った１６文字の文字列で表される。chunkgroup_time_base_flagsには、chunkgroupの基準カウンタに関するflagが記録される。chunkgroup_time_base_offsetには、chunkgroup内の基準時間軸の開始時刻が記録される。この値は、９０kHzのクロックをカウントアップするカウンタにセットされる値であり、３２bitで表される。text_block()は、さまざまなtextを格納するための領域であり、そのtext_block()で使用が許されているtext itemだけが記述される。

【０１２８】

図２２に示すように、chunk_connection_info()は、特異な点の情報（videoの切り替え点、videoとaudioの同期など）を記録しておくためのファイルであり、chunk間の接続状況を規定している。編集によりできたchunkとchunkのつなぎ目のような特異点では、GOPの途中でchunkを乗りかえる必要がある。このような編集点付近の情報がここに記述される。chunkは２つ以上のchunkgroupに属することはない。

【０１２９】

chunk_connection_info_lengthには、chunk_connection_info()の長さをbyte単位で表したものが記録される。number_of_chunksには、そのchunkgroupで使われるchunkの総数が記録される。chunk_sync_play_flagは、図３９に示すように、同時刻に２つ以上のchunkを再生する必要があるかどうかを表すflagであり、その値の０は、１つのchunkの再生を意味し、その値の１は、複数のchunkの同時再生を意味する。

【０１３０】

図２３のchunk_arrangement_info()において、chunk_arrangement_info_lengthには、

各chunkについての情報の長さをbyte単位で表したもの（chunk_arrangement_info_lengthの先頭byteからtransition_info()の最終byteまでを含めた長さ）が記録される。chunk_info_file_idには、対象となるchunk_info_fileのfile_idが記録される。

【０１３１】

chunk_switch_stream_idには、接続したとき連続再生するstreamのstream_idが記録される。このidとしては、例えば、MPEG2のパケットヘッダに記録されているビデオあるいはオーディオを識別するidが用いられる。presentation_start_cg_time_countには、該当chunkの表示開始時刻を、chunkgroup内の時刻で表したtime count値が記録される。chunkの表示開始時刻は、chunkgroup内で定義されるglobalなtime stampで表現される。該当chunkは、chunkgroupにおいて、この時刻から表示が開始される。presentation_end_cg_countには、該当chunkの表示終了時刻を、chunkgroup内の時刻で表したtime count値が記録される。chunkの表示終了時刻は、chunkgroup内で定義されるglobalなtime stampで表現される。

【０１３２】

図４０に示すように、original_time_count_typeには、stream内部で使われているtime countの種類が記録される。例えば、MPEG2 videoのstreamであれば、original_time_count_typeは’００００’とされる。number_of_start_original_time_count_extensionには、複数のtime countが必要なとき、新たに必要な開始時刻を表すtime countの数が記録される。number_of_end_original_time_count_extensionには、複数のtime countが必要なとき、新たに必要な終了時刻を表すtime countの数が記録される。presentation_start_original_time_countには、presentation_start_tg_time_countに対応する、stream内部における時刻あるいはカウンタ値が記録される。presentation_end_original_time_countには、presentation_end_tg_time_countに対応する、stream内部における時刻あるいはカウンタ値が記録される。

【０１３３】

tc_ext_attributesには、time_count_extensionに対する属性が記録される。このtime_count_extensionには、例えば、どのstreamに適用するのか等の情報を入れることができる。start_original_time_count_extensionには、chunkの切り替えに必要な開始時刻あるいは開始カウンタ値が記録される。これはオプションであり、複数の時刻やカウンタ値を記録する必要があるときに使用される。end_original_time_count_extensionには、chunkの切り替えに必要な終了時刻あるいは終了カウンタ値が記録される。これもオプションであり、複数の時刻やカウンタ値を記録する必要があるときに使用される。transition_info()には、chunkの切り替えで特殊効果をかけるときに必要な情報が記録される。例えば、chunkの指定、切り替え時刻、特殊効果の種類等がここに記述される。

【０１３４】

図２４に示すように、CHUNK_%%%%.ABSTは、sub_file番号%%%%のchunkを構成するbitstreamから抽出した特徴点を記録したfileである。このfileには、GOP,Audio frame等のbitstreamを構成する単位毎に、その開始byte位置、長さ、属性等が記述される。GOP情報、Audio frame情報はchunk(sub-file)毎に、１つのCHUNK_%%%%.ABSTとしてまとめられる。

【０１３５】

図４１に示すように、CHUNK_%%%%.ABSTのfile_type_idには、stream_info()が記録されているfileであることを示す識別子が、ISO 646に従った１６文字の文字列で記録される。

【０１３６】

図４２に示すように、info_typeには、図２４において次に続くstream_infoのtypeが記録される。ここでstreamの種類が特定される。number_of_programsには、MPEG2のTS(Transport Stream)に含まれるprogramの数が記録される。この数を知るには、PSI(Program Specific Information)を読み取る必要がある。TS以外のときには、この値は１になる。number_of_streamsには、そのprogramで使われるストリームの数が記録される。この値は、TSの場合には、異なるPID(packet identification)の数と等しくなる。TS以外のMPEGstrea

mの場合、ここには、stream idが異なるstreamの数が記録される。

【０１３７】

stream_identifierには、stream idが記録される。TSの場合には、stream idとして、PIDが利用される。

【０１３８】

図４３に示すように、slot_unit_typeには、streamをある一定間隔毎に区切ったときの、その区切り方が記録される。各frame, field等、区切りの指標が時間の場合には、time stamp valueが用いられる。slot_time_lengthには、１slotに対応する時間が記録される。この値は、９０kHzのクロックをカウントするカウンタを用いたtime stampの値で表される。number_of_slotsには、CHUNK_%%%%.ABSTに書かれているslot_info()の数が記録される。number_of_I_pictures_in_a_slotには、slotに含まれるI-pictureの数が記録される。この値は、１以上の整数で１５以下の値となる。ただし、GOPheaderを先頭とするslotの直前に位置するslotに含まれるI-pictureの数は、この値よりも小さくてもよい。GOPheaderの直後でないI-pictureのpicture headerを先頭とするslotを設定するとき、この値が活用される。

【０１３９】

次に、図１７と図１８に示したprogramについてさらに説明する。PROGRAM_$$$.PGIには、program()がただ１つ存在する。volume内にはprogramの数だけPROGRAM_$$$.PGIが存在する。program番号は、program()の中で定義せず、file名またはfile idで規定される。

【０１４０】

図４４に示すように、図１７のPROGRAM_$$$.PGIのfile_type_idには、program()が記録されたfileであることを示すidが、長さ１６の文字列で記録される。text_block()には、さまざまなtextを格納するための領域が形成されている。ここには、そのtext_block()で使用が許されているtext itemだけが記述される。

【０１４１】

図１８のprogram()のflags_for_programには、programに関する各種フラグが記録される。例えば、このprogramの書き込み属性、書き込み（変更許可）、再生回数制限、ratingのlevel等が記録される。

【０１４２】

図４５に示すように、program_statusには、programの属性が記録される。このfieldの設定はoptionであるが、設定しないときは"none"にしなければならない。

【０１４３】

program_playback_time()には、そのprogramの再生時間が記録される。number_of_play_sequencesには、そのprogramで使用されているplay_sequenceの数が記録される。ただし、このformatの例では、値は１に固定されている。すなわち、このformatの例では、１ program＝１ch（チャンネル）再生とされているので、２ch同時再生を実現するには、２programの同時再生指定を可能にすればよい。１program＝１ch再生の制限がなければ、１programで、２ch同時再生も可能である。multichannel Ｉ／Ｏを使って２つのplay sequenceを同時再生するとき、どのplay sequenceをどのoutput channelに割り当てるかは、光ディスク装置が決める。

【０１４４】

number_of_play_listsには、このplay sequenceで使用されているplay_listの数が記録される。この例では値は１とされる。play_list_start_time_stamp_offsetには、play sequenceの開始時刻から開始するtimerでカウントした、play sequence内での時刻が記録される。この値が、play listの開始時刻になる。programでは、play sequence内にplay listは１つしか存在してはならない。時刻の単位系は９０kHzである（１／９００００秒が時刻の最小単位である）。stuffing_bytesには、stuffingのbytesが記録される。その長さは、８×n bit(n>=０）と
される。

【０１４５】

次に、不連続点フラグについてさらに説明する。ここで不連続点フラグとは、図１９のplay_list()のseamless_connection_flag、または図３７に示したindex type 8として記録されるマークを意味する。

【０１４６】

seamless_connection_flagの値の０は、図４６に示すように、直前のplay itemとの連続再生（シームレス再生）が保証されていないこと、または不明であることを意味し、その値の１は、シームレス再生が保証されていることを意味する。

【０１４７】

すなわち、図４７に示すように、このflagが０である場合には、所定のplay itemが直前のplay itemから連続して（画像あるいは音声を途切れさせることなく）再生することが保証されている。これに対して、このflagが１である場合には、直前のplay itemが終了した後、次のplay itemが再生されるまでの間に不連続部分が発生する場合があることになる。

【０１４８】

次に、図４８のフローチャートを参照して、titleの不連続点フラグを記録する処理を説明する。最初にステップＳ１において、ユーザは、入力部１４を操作して、記録対象とするtitleを指定する。このとき、ステップＳ２において、CPU２１は、光ディスク１にステップＳ１で指定されたtitleを記録したとき、そのtitleを、既に記録されている直前のtitleに対して連続して再生することが可能であるか否かを判定する。この判定は、読み出しチャネル用バッファ６の容量（その容量のデータを、デコーダ７がデコードするのに要する時間）と、title間の再生時間に対応して行われる。すなわち、次のtitleを再生するまでの間に、読み出しチャネル用バッファ６が空になってしまうときは、連続再生が不可能と判定され、空にならないときは、連続再生が可能と判定される。

【０１４９】

連続再生が可能でない場合には、ステップＳ３に進み、CPU２１は、OSD制御回路９を制御し、いま記録が指令されたtitleは、既に記録されているtitleに対して、連続再生することができないこと、すなわち、両者の間には、不連続点が発生することを表すメッセージを発生させる。このメッセージは、合成回路８から、出力端子Ｐ１を介して、ディスプレイに表示される。

【０１５０】

ユーザは、このメッセージを見て、不連続点の発生を了承するか否かを入力部１４を操作して入力する。ユーザが不連続点の発生を了承した場合、CPU２１は、ステップＳ５において、図１６のtitle_info()中のmark_typeに、図３７に示すマークのうちの、不連続点を示すインデックスとしてのindex type 8を設定させるとともに、不連続点を発生させる位置を、図１６のrelative_time_stamp_in_titleに設定する処理を実行する。なお、このrelative_time_stamp_in_titleに記録する位置（不連続点を発生させる位置）は、任意の位置とすることができる。従って、通常、不連続点が発生しても影響が少ない点（例えば、次のtitleの
先頭、または、前のtitleの最後）がここに記録される。

【０１５１】

そして、このような設定が行われたtitle_info()は、RAM２４から書き込みチャネル用バッファ１１に供給され、記憶された後、所定のタイミングで、そこから読み出され、スイッチ５、ECC回路４、ＲＦおよび復調／変調回路３、および光ヘッド２を介して、光ディスク１に供給され、記録される。

【０１５２】

次に、ステップＳ６に進み、CPU２１は、title内に他にも連続して再生することができない点が存在するか否かを判定し、再生不連続点が他にも存在する場合には、ステップＳ３に戻り、それ以降の処理を繰り返し実行する。title内に再生不連続点が他には存在しないと判定された場合、処理は終了される。

【０１５３】

一方、ステップＳ４において、ユーザが不連続点の発生を了承しない旨を入力した場合、ステップＳ７に進み、CPU２１は、既に記録済みのtitleの記録位置を、これから記録するtitleと連続再生可能な位置に変更可能か否かを判定する。titleの記録位置を変更可能であると判定した場合、CPU２１は、ステップＳ８に進み、既に記録されているtitleの記録位置を、これから記録するtitleと連続再生可能な位置に変更する。その後、ステップＳ６に進む。

【０１５４】

ステップＳ７において、titleの記録位置を変更することができないと判定された場合、ステップＳ５に進み、上述したように、mark_typeとして不連続点が記録されるとともに、不連続点の発生位置がrelative_time_stamp_in_titleに記録される。

【０１５５】

このような処理の後、ステップＳ１で指定されたtitleの記録処理が実行される。

【０１５６】

次に、図４９のフローチャートを参照して、program作成時に不連続点フラグを記録する場合の処理について説明する。最初にステップＳ２１において、ユーザは、入力部１４を操作して、programに含めるtitleを指定する。この指定が行われたとき、ステップＳ２２において、CPU２１は、ステップＳ２１で指定されたtitleの中の必要な部分を、再生開始点および再生終了点で指定する。CPU２１は、この指定に対応して、play item（図２０）を作成する。

【０１５７】

次に、ステップＳ２３において、CPU２１は、直前のplay itemとの間で連続再生が可能か否かを判定し（この判定も、play itemとplay itemの間に発生する再生時間と、読み出しチャネル用バッファ６の容量（その容量のデコード時間）との比較に基づいて行われる）、連続再生が可能でない場合には、ステップＳ２４に進み、OSD制御回路９を制御し、不連続点が発生することをメッセージで表示させる。このメッセージに対して、ユーザは、入力部１４を操作して、不連続点の発生を了承するか否かを入力する。そこで、CPU２１は、ステップＳ２５において、ユーザが不連続点の発生を了承したか否かを判定し、ユーザが不連続点の発生を了承したと判定した場合には、ステップＳ２６に進み、そのplay itemのseamless_connection_flagを０に設定する。図４６を参照して説明したように、このflagが０であるということは、連続再生が保証されていないことを意味する。

【０１５８】

次に、ステップＳ２７に進み、CPU２１は、programの作成処理が終了したか否かを判定し、まだ終了していないと判定した場合には、ステップＳ２３に戻り、それ以降の処理を繰り返し実行する。ステップＳ２７において、programの作成処理を終了したと判定した場合、処理は終了される。

【０１５９】

一方、ステップＳ２５において、ユーザが、不連続点の発生を了承しないと判定された場合、ステップＳ２８に進み、CPU２１は、対象としているストリームの光ディスク１上の記録位置（配置位置）を変更するか、あるいは部分的再記録が可能であるか否かを判定する。ストリームの再配置処理または部分的再記録が可能である場合には、ステップＳ２９に進み、CPU２１は、記録位置を連続再生可能な位置に変更する処理を実行する。そして、この場合には、ステップＳ３０において、CPU２１は、そのplay itemのseamless_connection_flagを１に設定する。図４６を参照して説明したように、このflagの１は、連続再生が保証されていることを意味する。その後、ステップＳ２７に進み、それ以降の処理が実行される。

【０１６０】

ステップＳ２８において、ストリームの再配置処理または部分的再記録が不可能であると判定された場合には、ステップＳ２６に進み、ユーザが不連続点の発生を了承した場合と同様の処理が実行される。

【０１６１】

ステップＳ２３において、直前のplay itemとの間で連続再生が可能であると判定された場合には、ステップＳ３０に進み、直ちに、そのplay itemのseamless_connection_flagに１が設定される。

【０１６２】

次に、図４８を参照して説明したように、titleに不連続点フラグが記録された場合に、その再生が指令されたときの処理について、図５０のフローチャートを参照して説明する。最初に、ステップＳ４１において、CPU２１は、指定されたtitleの不連続点フラグを読み取る。ステップＳ４２において、CPU２１は、titleの再生処理を開始し、ステップＳ４３において、再生を開始したtitleが終了したか否かを判定する。titleの再生が終了していない場合には、ステップＳ４４に進み、CPU２１は、不連続点フラグ（relative_time_stamp_in_title）で表された位置が再生されたか否かを判定し、まだ再生されていないと判定された場合には、ステップＳ４３に戻る。

【０１６３】

ステップＳ４４において、不連続点フラグで表された位置が再生されたと判定された場合、ステップＳ４５に進み、CPU２１は、プレーヤ（いまの場合、光ディスク装置）が後続部分の連続再生のため、ギャップが必要であるか否かを判定する。この判定は、読み出しチャネル用バッファ６に、その時点において記憶されているデータ量（そのデータ量をデコーダ７がデコードするのに要する時間）に基づいて行われる。そのデータ量が充分である場合には、ギャップは不要と判定され、そのデータ量が不充分である場合には、ギャップが必要と判定される。この光ディスク装置が後続部分の連続再生のためギャップを必要としていない場合（読み出しチャネル用バッファ６にデータが充分記憶されている場合）には、ステップＳ４３に戻り、それ以降の処理が繰り返し実行される。

【０１６４】

これに対して、ステップＳ４５において、光ディスク装置が後続部分の再生のためにギャップを必要としていると判定された場合（読み出しチャネル用バッファ６にデータが充分記憶されていない場合）、ステップＳ４６に進み、CPU２１は、ギャップを発生させる。すなわち、CPU２１は、読み出しチャネル用バッファ６に対するデータの書き込みはそのまま継続させるが、デコーダ７に対するその読み出しは中断させ、読み出しチャネル用バッファ６のデータ量を増加させる。そして、ステップＳ４７に進み、後続部分の連続再生が可能になったか否かを判定し、まだ可能になっていない場合には、ステップＳ４６に戻り、ギャップ発生処理を継続させる。

【０１６５】

以上のようにして、所定量のデータが読み出しチャネル用バッファ６に書き込まれ、ステップＳ４７において、後続部分の連続再生が可能な状態になったと判定された場合、ステップＳ４８に進み、CPU２１は、再び再生を継続（デコーダ７によるデコードを再開）させる。その後、ステップＳ４３に戻り、それ以降の処理が繰り返し実行される。

【０１６６】

一方、ステップＳ４３において、titleが終了したと判定された場合、ステップＳ４９に進み、次に再生するtitleが存在するか否かをCPU２１は判定する。次に再生するtitleが存在しない場合には、処理は終了される。これに対して、次に再生するtitleが存在すると判定された場合には、ステップＳ５０に進み、CPU２１は、前のtitleの最後に不連続フラグが存在するか否かを判定する。前に再生したtitleの最後に不連続フラグが存在すると判定された場合、ステップＳ５１に進み、CPU２１は、プレーヤ（光ディスク装置）は次のtitleの再生のためにギャップを必要とするか否かを判定する。ギャップが必要であると判定された場合、ステップＳ５２に進み、CPU２１は、ギャップ発生処理を実行し、ステップＳ５３において、次のtitleの再生が可能になったか否かを判定し、可能になっていない場合には、ステップＳ５２に戻り、ギャップ発生処理を繰り返し実行する。

【０１６７】

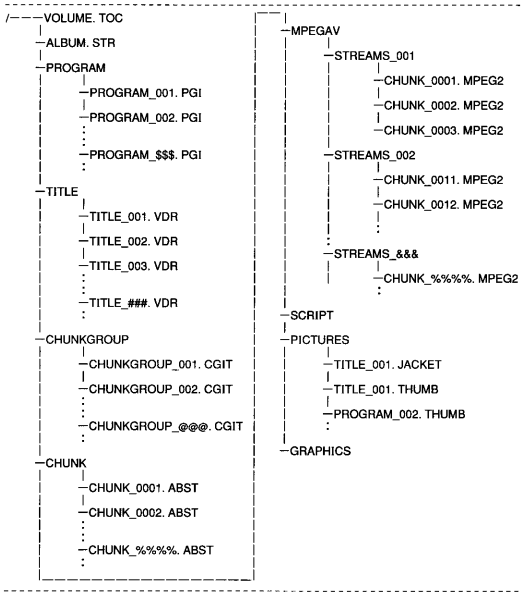
ステップＳ５３において、次のtitleの再生が可能になったと判定された場合、ステップＳ４１に戻り、それ以降の処理が繰り返し実行される。

【０１６８】
ステップＳ５０において、前のtitleの最後に不連続フラグが存在しないと判定された場合、ステップＳ４１に戻り、それ以降の処理が実行される。また、前のtitleの最後に不連続フラグが存在したとしても、ステップＳ５１において、光ディスク装置が次のtitleの再生のためにギャップを必要としないと判定された場合（その光ディスク装置の読み出しチャネル用バッファ６の容量が充分大きい場合）には、ステップＳ４１に戻り、それ以降の処理が実行される。
【０１６９】
次に、図４９に示すようにして、programが作成された場合に、そのprogramの再生が指令されたときの処理について、図５１のフローチャートを参照して説明する。
【０１７０】
最初に、ステップＳ６１において、CPU２１は、指定されたprogramを構成する最初のplay itemを再生する。ステップＳ６２において、CPU２１は、play itemの再生が終了したか否かを判定し、まだ終了していない場合には、終了するまで待機する。
【０１７１】
ステップＳ６２において、play itemの再生が終了したと判定された場合、CPU２１は、ステップＳ６３に進み、次のplay itemが存在するか否かを判定する。次のplay itemが存在すると判定された場合、ステップＳ６４に進み、CPU２１は、次のplay itemのseamless_connection_flagが０であるか否かを判定する。このflagが０であると判定された場合（シームレス再生が保証されていない場合）、ステップＳ６５に進み、CPU２１は、光ディスク装置が次のplay item再生のためにギャップを必要とするか否かを判定し、ギャップを必要とする場合には、ステップＳ６６に進み、ギャップ発生処理を実行する。そして、ステップＳ６７において、次のplay itemを再生することが可能な状態になったか否かを判定し、
まだ可能な状態になっていない場合には、ステップＳ６６に戻り、ギャップ発生処理を繰り返し実行する。
【０１７２】
ステップＳ６７において、次のplay itemを再生することが可能な状態になったと判定された場合（読み出しチャネル用バッファ６に充分な量のデータが書き込まれた場合）、ステップＳ６８に進み、CPU２１は、次のplay itemの再生処理を開始する。その後、ステップＳ６２に戻り、それ以降の処理が繰り返し実行される。
【０１７３】
ステップＳ６４において、次のplay itemのseamless_connection_flagが０ではないと判定された場合（１であると判定された場合）、あるいはまた、ステップＳ６５において、光ディスク装置が次のplay item再生のためにギャップを必要としていないと判定された場合、ステップＳ６６とステップＳ６７の処理はスキップされ、ステップＳ６８に進む。そして、次のplay itemの再生が開始され、その後、ステップＳ６２に戻る。
【０１７４】
以上においては、本発明を光ディスク装置に応用した場合を例として説明したが、本発明は、その他の記録媒体に情報を記録または再生する場合にも適用することが可能である。
【０１７５】
なお、上記したような処理を行うコンピュータプログラムをユーザに提供する提供媒体としては、磁気ディスク、CD-ROM、固体メモリなどの記録媒体の他、ネットワーク、衛星などの通信媒体を利用することができる。
【図面の簡単な説明】
【０１７６】
【図１】ディレクトリの構造を説明する図である。
【図２】VOLUME.TOCを説明する図である。
【図３】volume_information()を説明する図である。

【図４】volume_attribute()を説明する図である。
【図５】resume()を説明する図である。
【図６】volume_rating()を説明する図である。
【図７】write_protect()を説明する図である。
【図８】play_protect()を説明する図である。
【図９】recording_timer()を説明する図である。
【図１０】text_block()を説明する図である。
【図１１】language_set()を説明する図である。
【図１２】text_item()を説明する図である。
【図１３】ALBUM.STRを説明する図である。
【図１４】album()を説明する図である。
【図１５】TITLE_###.VDRを説明する図である。
【図１６】title_info()を説明する図である。
【図１７】PROGRAM_$$$.PGIを説明する図である。
【図１８】program()を説明する図である。
【図１９】play_list()を説明する図である。
【図２０】play_item()を説明する図である。
【図２１】CHUNKGROUP_###.CGITを説明する図である。
【図２２】chunk_connection_info()を説明する図である。
【図２３】chunk_arrangement_info()を説明する図である。
【図２４】CHUNK_%%%%.ABSTを説明する図である。
【図２５】本発明を適用した光ディスク装置の構成例を示すブロック図である。
【図２６】ディレクトリの構造を説明する図である。
【図２７】ディレクトリの論理構造を説明する図である。
【図２８】offsetを説明する図である。
【図２９】ディレクトリの構造を説明する図である。
【図３０】ディレクトリの構造を説明する図である。
【図３１】ディレクトリの論理構造を説明する図である。
【図３２】ディレクトリの構造を説明する図である。
【図３３】ディレクトリの構造を説明する図である。
【図３４】ディレクトリの論理構造を説明する図である。
【図３５】ディレクトリの論理構造を説明する図である。
【図３６】file_type_idを説明する図である。
【図３７】mark_typeを説明する図である。
【図３８】file_type_idを説明する図である。
【図３９】chunk_sync_play_flagを説明する図である。
【図４０】original_time_count_typeを説明する図である。
【図４１】file_type_idを説明する図である。
【図４２】info_typeを説明する図である。
【図４３】slot_unit_typeを説明する図である。
【図４４】file_type_idを説明する図である。
【図４５】program_statusを説明する図である。
【図４６】seamless_connection_flagを説明する図である。
【図４７】seamless_connection_flagの意味を説明する図である。
【図４８】titleの不連続点フラグ記録処理を説明するフローチャートである。
【図４９】program作成時の不連続点フラグの記録処理を説明するフローチャートである。
【図５０】title再生時の不連続点フラグの処理を説明するフローチャートである。
【図５１】program再生時の不連続点フラグの処理を説明するフローチャートである。
【符号の説明】

【０１７７】
１　光ディスク，　２　光ヘッド，　３　ＲＦおよび復調／変調回路，　４　ECC回路，　６　読み出しチャネル用バッファ，　７　デコーダ，　８　合成回路，　９　OSD制御回路，　１０　エンコーダ，　１１　書き込みチャネル用バッファ，　１２　アドレス検出回路，　１３　システムコントローラ，　１４　入力部，　２１　CPU，　２２　ROM，　２３，２４　RAM

【図１】



【図２】

Syntax	Number of Bits	Mnemonic
VOLUME.TOC {		
file_type_id	8*16	char[16]
volume_information ()		
text_block ()		
}		

【図３】

Syntax	Number of Bits	Mnemonic
volume_information () {		
volume_attribute ()		
resume ()		
volume_rating ()		
write_protect ()		
play_protect ()		
recording_timer ()		
}		

【図４】

Syntax	Number of Bits	Mnemonic
volume_attribute () {		
volume_attribute_length	32	uimsbf
vdr_version	4*4	bcd
reserved	6	bslbf
title_playback_mode_flag	1	bslbf
program_playback_mode_flag	1	bslbf
volume_play_time ()	4*8	bcd
update_time_count ()	32	uimsbf
maker_id	8*16	char[16]
model_code	8*16	char[16]
POSID	32	bslbf
}		

【図５】

Syntax	Number of Bits	Mnemonic
resume () {		
resume_length	32	uimsbf
reserved // for byte alignment	3	bslbf
resume_switch	1	bit
reserved	4	bslbf
number_of_records	4	uimsbf
reserved // for byte alignment	7	bslbf
resume_auto_execute_time_flag	1	bit
resume_auto_execute_time ()	4*14	bcd
reserved	4	bslbf
resume_auto_execute_record_number	4	uimsbf
for (i=0; i<number_of_records; i++) {		
resume_mode_flag	4	bslbf
object_type	4	bslbf
linked_record_number	4	uimsbf
number_of_times	16	uimsbf
resume_updated_time ()	4*14	bcd
switch (object_type) {		
case title :		
title_number	16	uimsbf
title_local_time_stamp	64	uimsbf
break ;		
case program :		
program_number	16	uimsbf
program_local_time_stamp	64	uimsbf
break ;		
case program_bind :		
program_bind_number	16	uimsbf
program_order	16	uimsbf
program_number	16	uimsbf
program_local_time_stamp	64	uimsbf
break ;		
case play_item :		
play_item_number	16	uimsbf
play_item_local_time_stamp	64	uimsbf
break		
}		
}		
}		

【図６】

Syntax	Number of Bits	Mnemonic
volume_rating () {		
volume_rating_length	32	uimsbf
reserved	6	bslbf
volume_rating_type	2	bslbf
volume_rating_password	128	bslbf
switch (rating_type) {		
case age_limited :		
country_code_for_rating	32	bslbf
for (i=0; i<32; i++) {		
rating_bit_for_age_limited	1	bslbf
break ;		
case CARA :		
CARA_category	4	bslbf
reserved	4	bslbf
reserved	16	bslbf
break ;		
case RSAC :		
RSAC_category	4	bslbf
level	4	bslbf
reserved	16	bslbf
break ;		
}		
}		

【図７】

Syntax	Number of Bits	Mnemonic
write_protect () {		
write_protect_length	32	uimsbf
volume_write_protect_level	4	uimsbf
password_enable_flag	1	bslbf
append_only_flag	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_write_protect	128	bslbf
reserved	8	bslbf
write_protect_set_time ()	56	bcd
reserved	8	bslbf
write_protect_expiration_time ()	56	bcd
number_of_times	16	uimsbf
}		

【図８】

Syntax	Number of Bits	Mnemonic
play_protect () {		
play_protect_length	32	uimsbf
volume_play_protect_flag	2	bslbf
reserved	2	bslbf
password_enable_flag	1	bslbf
reserved	1	bslbf
expiration_time_enable_flag	1	bslbf
number_of_times_enable_flag	1	bslbf
password_for_volume_play_protect	128	bslbf
reserved	8	bslbf
play_protect_set_time ()	56	bcd
reserved	8	bslbf
play_protect_expiration_time ()	56	bcd
number_of_times	16	uimsbf
}		

【図１０】

Syntax	Number of Bits	Mnemonic
text_block () {		
text_block_length	32	uimsbf
number_of_language_sets	8	uimsbf
number_of_text_items	16	uimsbf
for (i=0; i<number_of_language_sets; i++) {		
language_set ()		
}		
for (i=0; i<number_of_text_items; i++) {		
text_item ()		
}		
}		

【図９】

Syntax	Number of Bits	Mnemonic
recording_timer () {		
recording_timer_length		
recording_timer_flag		
number_of_entry		
for (i=0; i<number_of_entry; i++) {		
date_and_time		
channel		
program		
:		
}		
}		

【図１１】

Syntax	Number of Bits	Mnemonic
language_set () {		
reserved	8	bslbf
language_code	24	bslbf
number_of_language_set_names	8	uimsbf
for (i=0; i<number_of_language_set_names; i++) {		
character_set_type	8	bslbf
language_set_name_length	8	uimsbf
language_set_name	8*language_set_name_length	bslbf
}		
}		

【図１２】

Syntax	Number of Bits	Mnemonic
text_item () {		
text_item_length	16	uimsbf
text_item_id	16	uimsbf
text_item_sub_id	16	uimsbf
flags	8	bslbf
number_of_used_language_sets	8	uimsbf
// loop for each language set		
for (i=0; i<number_of_used_language_sets; i++) {		
language_set_id	8	uimsbf
reserved	4	bslbf
text_string_length	16	uimsbf
text_string	8*text_string_length	bslbf
bitman ()		
}		
stuffing_bytes	8*n	bslbf
}		

【図１３】

Syntax	Number of Bits	Mnemonic
ALBUM. STR {		
file_type_id	8*16	char[16]
album ()		
text_block ()		
}		

【図１４】

Syntax	Number of Bits	Mnemonic
album () {		
album_length	32	uimsbf
reserved	6	bslbf
volume_status	1	bslbf
if (volume_status== "1b") {		
chief_volume_flag	1	bslbf
} else {		
reserved	1	"0"
}		
if (volume_status== "1b") {		
if (chief_volume_flag== "1b") {		
reserved	6	bslbf
album_type	2	bslbf
albim_id	128	bslbf
number_of_discs_in_album	16	uimsbf
number_of_volumes_in_album	16	uimsbf
for (i=0; i<number_of_volumes_in_album; i++) {		
disc_id_for_album_member	128	bslbf
volume_id_for_album_member	128	bslbf
title_offset_number	16	uimsbf
}		
reserved_for_program_bind	8	bslbf
number_of_program_binds	8	uimsbf
for (i=0; i<number_of_program_binds; i++) {		
number_of_program_in_this_program_bind	16	uimsbf
for (i=0; i<number_of_programs_in_this_program_binds; i++) {		
disc_id_for_program_bind_member	128	uimsbf
volume_id_for_program_bind_member	128	uimsbf
program_number	16	uimsbf
}		
} else { // chief_volume_flag== "0b"		
chief_disc_id	128	uimsbf
chief_volume_id	128	uimsbf
(album_id	128	bslbf
}		
}		
}		

【図１５】

Syntax	Number of Bits	Mnemonic
TITLE_###. VDR {		
file_type_id	8*16	char[16]
title_info ()		
text_block ()		
}		

【図１６】

Syntax	Number of Bits	Mnemonic
title_info () {		
title_info_length	32	uimsbf
flags_for_title	32	bslbf
cgit_file_id	16	uimsbf
title_start_chunk_group_time_stamp	64	uimsbf
title_end_chunk_group_time_stamp	64	uimsbf
title_playback_time ()	32	bcd
reserved	32	bslbf
number_of_marks	16	uimsbf
for (i=0; i<number_of_marks; i++) {		
reserved	4	bslbf
mark_type	4	bslbf
relative_time_stamp_in_title	64	uimsbf
}		
stuffing_bytes	8*n	bslbf
}		

【図１７】

Syntax	Number of Bits	Mnemonic
PROGRAM_$$$. PGI {		
file_type_id	8*16	char[16]
program ()		
text_block ()		
}		

【図１８】

Syntax	Number of Bits	Mnemonic
program () {		
program_length	32	uimsbf
flags_for_program	32	bslbf
program_status	4	bslbf
program_playback_time ()	32	bcd
reserved	32	bslbf
number_of_play_sequences	16	uimsbf
for (j=0; j<number_of_play_sequence; j++) {		
number_of_play_lists	16	uimsbf
for (k=0; k<number_of_play_lists; k++) {		
play_list_start_time_stamp_offset	64	uimsbf
play_list (k)		
}		
}		
stuffing_bytes	8*n	bslbf
}		

【図１９】

Syntax	Number of Bits	Mnemonic
play_list () {		
// playback sequence of play items in this play list		
number_of_play_items	16	uimsbf
for (k=0; k<number_of_play_items; k++) {		
play_item_number	16	uimsbf
reserved	31	bslbf
seamless_connection_flag	1	bslbf
}		
// play_item_table		
for (PIN=1; PIN<=number_of_play_items_in_program; PIN++) {		
play_item ()		
}		
}		

【図２０】

Syntax	Number of Bits	Mnemonic
play_item () {		
play_item_length	32	uimsbf
play_item_type	8	bslbf
play_mode	8	bslbf
total_playback_time ()	32	bcd
menu_item_number	16	uimsbf
return_item_number	16	uimsbf
next_item_number	16	uimsbf
prev_item_number	16	uimsbf
if (play_item_type= "0000b") {		
// play item for one "cut"		
title_number	16	uimsbf
// IN point		
item_start_time_stamp	64	uimsbf
// OUT point		
item_end_time_stamp	64	uimsbf
}		
}		

【図２１】

Syntax	Number of Bits	Mnemonic
CHUNKGROUP_###. CGIT {		
file_type_id	8*16	char[16]
chunkgroup_time_base_flags	32	bslbf
chunkgroup_time_base_offset	64	uimsbf
chunk_connection_info ()		
text_block ()		
}		

【図２２】

Syntax	Number of Bits	Mnemonic
chunk_connection_info () {		
chunk_connection_info_length	32	uimsbf
reserved	16	bslbf
number_of_chunks	16	uimsbf
chunk_sync_play_flag	8	bslbf
// chunk info file list		
for (i=0; i<number_of_chunks; i++) {		
chunk_arrangement_info ()		
}		
}		

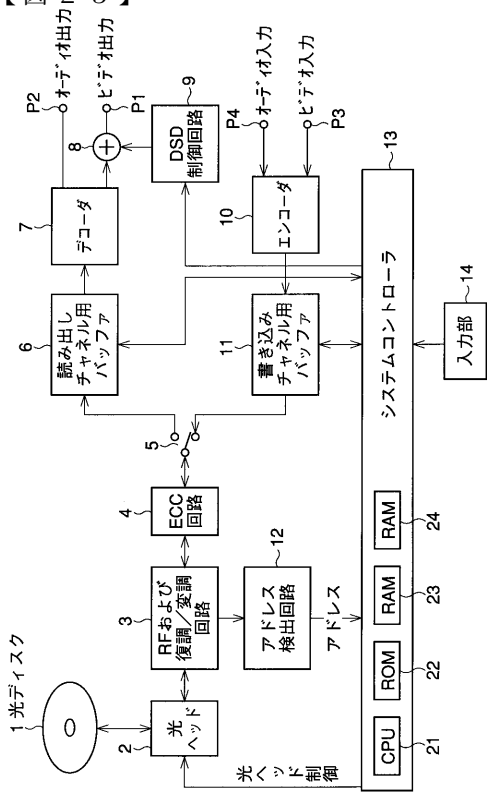
【図２３】

Syntax	Number of Bits	Mnemonic
chunk_arrangement_info () {		
chunk_arrangement_info_length	32	uimsbf
chunk_info_file_id	16	bslbf
reserved	5	bslbf
chunk_switch_stream_id	16	bslbf
presentation_start_cg_time_count	64	uimsbf
presentation_end_cg_time_count	64	uimsbf
reserved	4	bslbf
chunk_time_count_type	4	bslbf
number_of_start_original_time_count_extension	8	uimsbf
number_of_end_original_time_count_extension	8	uimsbf
// presentation start position and time		
presentation_start_original_time_count	64	uimsbf
presentation_end_original_time_count	64	uimsbf
for (j=0; j<number_of_start_original_time_count_extension; j++)		
tc_ext_attributes	16	bslbf
start_original_time_count_extension	64	uimsbf
}		
// presentation end position and time		
for (k=0; k<number_of_end_original_time_count_extension; k++) {		
tc_ext_attributes	16	bslbf
end_original_time_count_extension	64	uimsbf
}		
transition_info ()		
}		

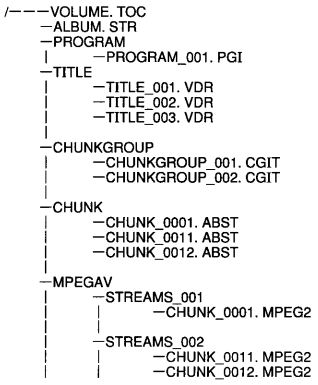
【図２４】

Syntax	Number of Bits	Mnemonic
CHUNK_%%%%. ABST {		
file_type_id	8*16	char[16]
reserved	4	bslbf
chunk_file_id	16	uimsbf
info_type	4	bslbf
// stream_info ()		
if (info_type== "MPEG2_System_TS") {		
number_of_programs	8	uimsbf
else {		
number_of_programs	8	"0000 0001"
}		
for (i=0; i<number<of_programs; i++) {		
number_of_streams	8	uimsbf
for (i=0; i<number_of_streams; i++) {		
stream_identifier	8	bslbf
// slot type information		
reserved	4	bslbf
slot_unit_type	4	bslbf
if (slot_unit_type== "time_stamp") {		
slot_time_length	32	uimsbf
} else {		
reserved	32	bslbf
}		
number_of_slots	32	uimsbf
reserved	4	bslbf
switch (info_type) {		
case MPEG1_System :		
case MPEG2_System_PS :		
case MPEG2_System_TS :		
case video_elementary_stream		
number_of_I_pictures_in_a_slot	4	uimsbf
break ;		
default :		
reserved	4	bslbf
break ;		
}		
// stream attribute		
ES_attribute ()		
}		
// loop of slot info		
for (i=0; i<number_of_streams; i++) {		
for (i=0; i<number_of_slots; i++) {		
slot_info ()		
}		
}		
}		
}		

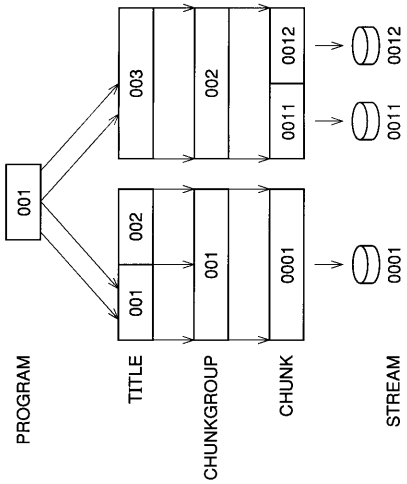
【図２５】



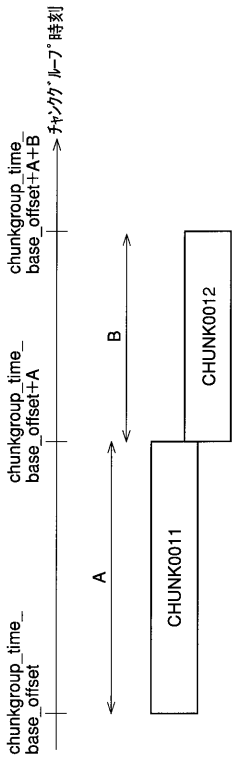
【図２６】



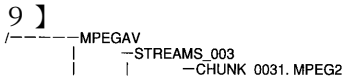
【図２７】



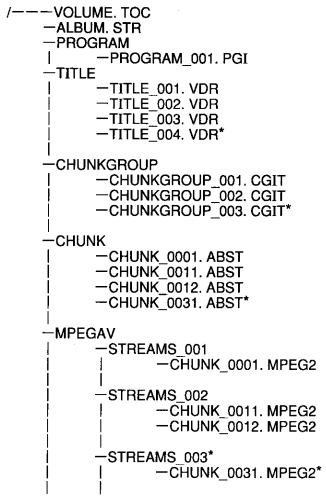
【図２８】



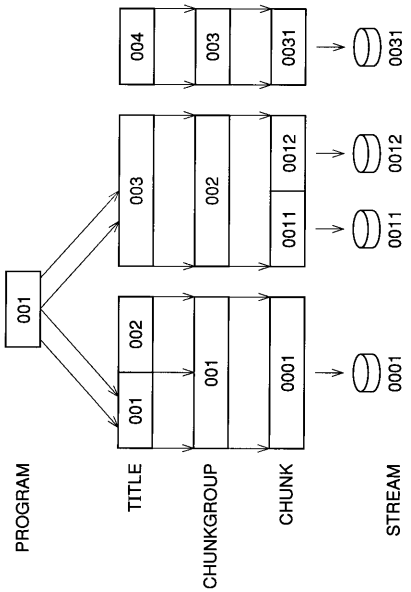
【図２９】



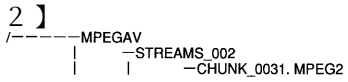
【図３０】



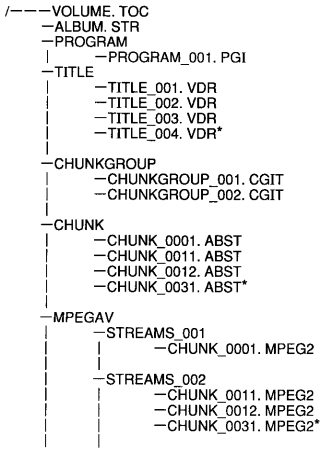
【図３１】



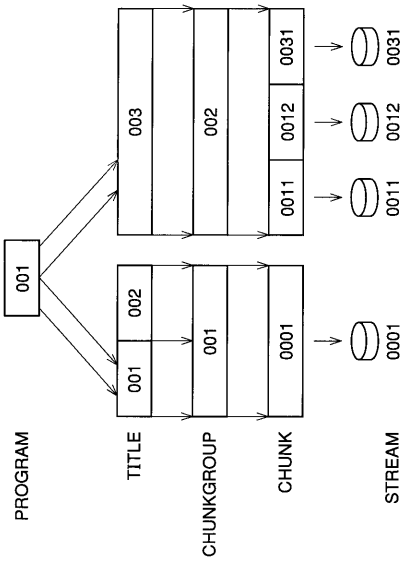
【図３２】



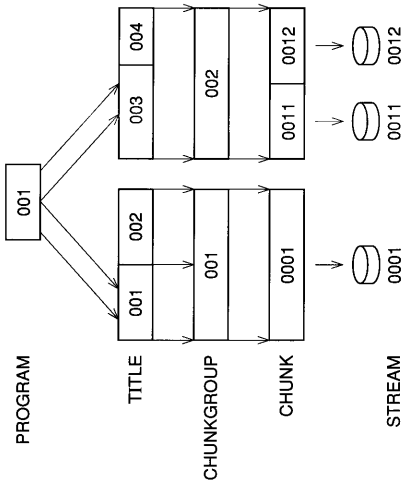
【図３３】



【図３４】



【図３５】



【図３６】

field_name	value
file_type_id	"TITLE_INFO_00000"

【図３８】

field_name	value
file_type_id	"CHUNKGROUP_0000"

【図３７】

mark_type	Meaning
0000b	index type 1
0001b	index type 2
0010b .. 0111b	index type 3 .. index type 8
1000b	skip IN
1001b	skip OUT
1010b	jump to title end or next title
1011b	scene change
1100b	audio mute
1101b	audio peak
1110b	picture still
1111b	reserved

index : direct entry point in the title
skip IN : start point of title skip
skip OUT : end point of title skip

【図３９】

chunk_sync_play_flag	Meaning
0b	play only one chunk at the same time
1b	need to play chunks simultaneously

【図４０】

original_time_count_type	Meaning
0000	MPEG2 System time stamp(90kHz)
0001	SMPTE timecode
0010	field(525/60)(59.94Hz)
0011	field(625/50)
0100	frame(525/60)(29.97Hz)
0101	frame(625/50)
0110	drop frame time code(525/60)
0111	non drop frame time code(525/60)
1000	60Hz
1001	50Hz
1010	24Hz
1011	second
1100 .. 1111	reserved

【図４１】

field_name	value
file_type_id	“STREAM_INFO_0000”

【図４２】

info_type	Meaning
0000	MPEG2_System_PS
0001	MPEG2_System_TS
0010	MPEG2_System_PES
0011	video_elementary_stream
0100	elementary_stream_except_video
0101	MPEG1_System_stream
0110	reserved
0111	reserved
1000	Consumer_DVC
1001 .. 1111	reserved

【図４３】

slot_unit_type	Meaning
0000b	‘time_stamp’ : time stamp value
0001b	‘GOP’ : one GOP(Group of pictures)
0010b	‘audio_frame’ : one audio frame
0011b .. 1111b	reserved

【図４４】

field_name	value
file_type_id	“PROGRAM_INFO_000”

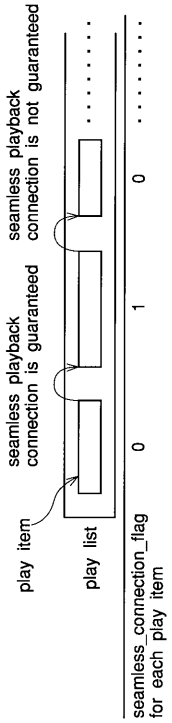
【図４５】

program_status	Meaning
0000b	none
0001b	original
0010b	copy
0011b	preview
0100b	rehearsal
0101b	temporary
0110b	complete
0111b	broken
1000b	editing
1001b	backup
1010b .. 1111b	reserved

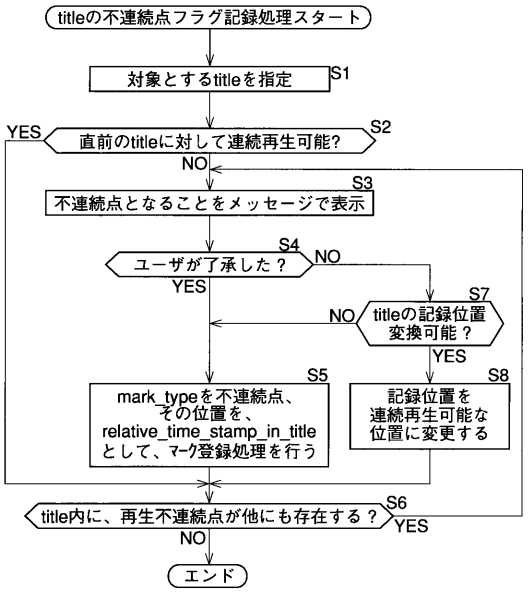
【図４６】

seamless_connection_flag	value
0b	seamless playback connection with the previous play item is not guaranteed or unknown
1b	seamless playback connection with the previous play item is guaranteed

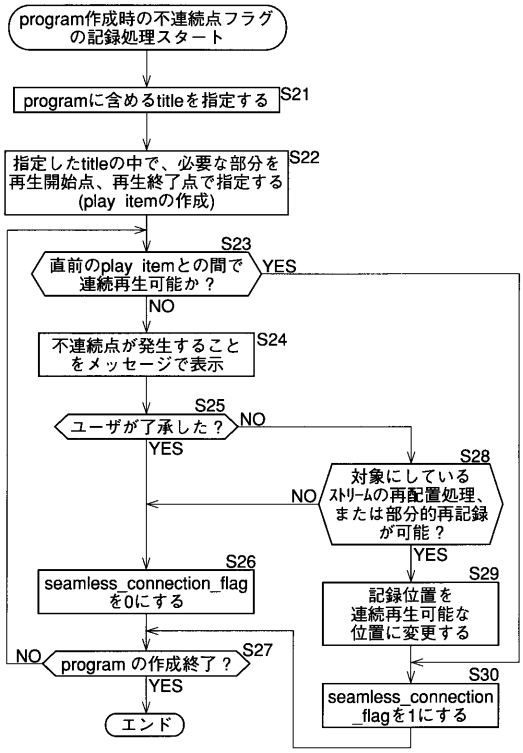
【図４７】



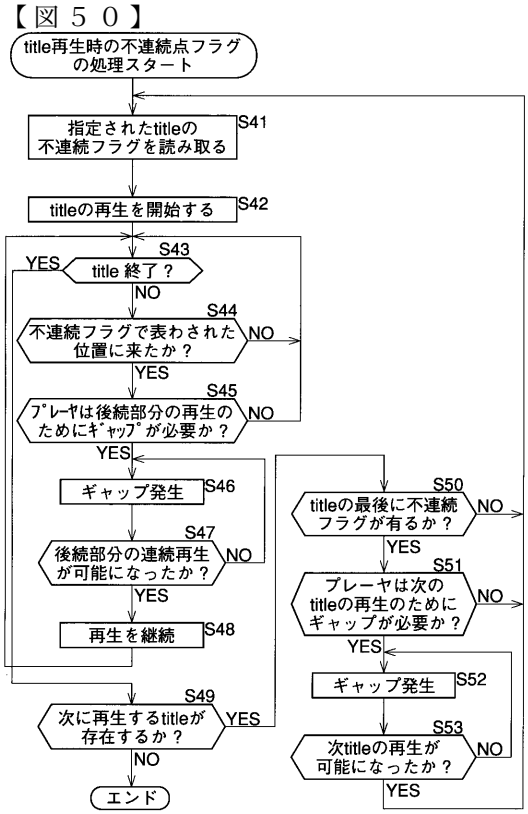
【図４８】



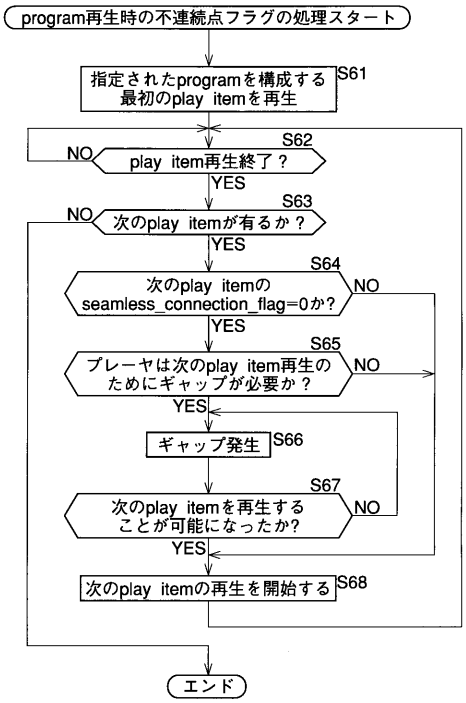
【図４９】



【図５０】



【図５１】


program再生時の不連続点フラグの処理スタート
指定されたprogramを構成する最初のplay itemを再生
S61
S62
play item再生終了？
NO
YES
S63
次のplay itemが有るか？
NO
YES
S64
次のplay itemのseamless_connection_flag=0か?
NO
YES
S65
プレーヤは次のplay item再生のためにギャップが必要か？
NO
YES
ギャップ発生
S66
S67
次のplay itemを再生することが可能になったか?
NO
YES
次のplay itemの再生を開始する
S68
エンド